PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/26, 29/06

A1

(11) International Publication Number: **WO 97/35406**

(43) International Publication Date: 25 September 1997 (25.09.97)

(21) International Application Number: PCT/US97/04658

(22) International Filing Date: 21 March 1997 (21.03.97)

(30) Priority Data:
08/620,263 22 March 1996 (22.03.96) US

(71) Applicant: TEKELEC CORPORATION [US/US]; 26580 West Agoura Road, Calabasas, CA 91302 (US).

(72) Inventors: SWETMAN, William, C.; 5305 Apartment H, Thicket Hill Lane, Raleigh, NC 27607 (US). BAKER, Jay; 6532 Wynbrook Way, Raleigh, NC 27612 (US).

(74) Agents: SMITH, Steven, W. et al.; Smith & Catlett, P.C., Suite 707, 2011 N. Collins Boulevard, Richardson, TX 75080-2645 (US).

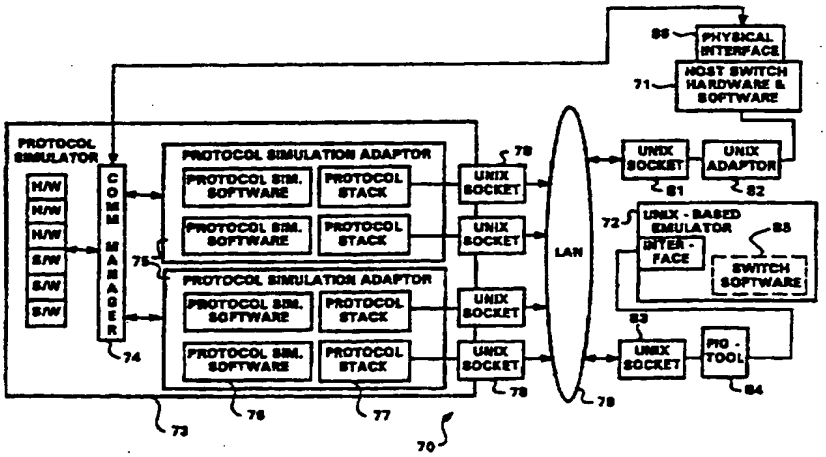
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: PROTOCOL SIMULATOR



(57) Abstract

A protocol simulator (73) for simulating open system interconnection (OSI) layers (3-7) of a plurality of communications protocols and utilizing the OSI layers to test selected ones of a plurality of target systems connected to a network (79). The protocol simuolator (73) includes a selection menu for selecting one of the plurality of target systems to be tested. The target system may be either a target hardware node (71) or an emulated target node (72). A plurality of protocol simulation software blocks (76) simulates OSI layers (3-7) of a plurality of different communications protocols, each of which is associated with one of the plurality of target systems. A protocol simulation adaptor (75) selects the protocol simulation software block (76) associated with the selected target system. The OSI layers of the selected protocol simulation software block are then packaged into a protocol stack (77), and test messages are generated utilizing the protocol stack. A UNIX application is then utilized to simulate OSI layers 1 and 2, and a UNIX socket (78) is utilized to transmit the test messages over the network (79) to the selected target system.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

BNSDOCID: <WO__9735406A1_I_>

PROTOCOL SIMULATOR

BACKGROUND OF THE INVENTION

Technical Field of the Invention

This invention relates to development test systems and, more particularly, to a protocol simulator and method of transmitting simulated open systems interconnect (OSI) protocol stacks over a network.

Description of Related Art

In the telecommunications industry, standard telecommunications networks are linked to each other using protocols based on the Open Systems Interconnection (OSI) model. The OSI model is an internationally accepted framework of standards for communication between different systems manufactured by different vendors. The OSI model creates an open systems networking environment where any vendor's computer system, connected to any network, freely shares data with any other computer system on that network or a linked network.

The OSI model organizes the communication process into seven different layers of interrelated protocols in a layered sequence based on their relation to the user. FIG. 1 is an illustrative block diagram of an OSI stack 10 illustrating the seven layers of the OSI model. Layers 1 through 3 deal with network access and layers 4 through 7 deal with end-to-end communications between the message source and the message destination. Each layer includes at least one function that is contained between an upper and a lower logical boundary. The services of each layer are combined with the services of lower layers to create new services that are made available to the higher layers. The layers are as follows:

Layer 1 is a physical layer that provides transmission of signals and the activation and deactivation of physical connections;

Layer 2 is a data link layer that includes signal synchronization, error correction, sequencing, and flow control. This layer also provides a data transmission link across one or several physical connections;

Layer 3 is a network layer that provides routing and switching functions;

Layer 4 is a transport layer utilizing layers 1 to 3 to provide an end-to-end service having required characteristics for the higher layer functions;

BNSDOCID: <WO 9735406A1 | >

Layer 5 is a session layer that provides the means to establish a session connection and to support an orderly exchange of data and related control functions for a particular communication service;

Layer 6 is a presentation layer that provides means for data formatting and code conversion; and

Layer 7 is an application layer, the protocols of which provide the actual service sought by an end user.

During development and testing of telecommunication systems and new service applications, it is common to develop a set of interrelated software programs which, when combined, model or simulate the system hardware. The effects of adding new service applications, or the effects of proposed hardware or software changes in the system, can be rapidly modeled and analyzed without undergoing the costly and time consuming process of actually modifying the system hardware or loading a new service application in an actual system. Problems arise, however, and the cost of testing increases substantially when the functions to be tested require communications between two or more telecommunication systems.

Existing communication links utilized for linking standard telecommunications systems generally comprise computing software that executes and assembles OSI layers 3-7 and transmission hardware that executes OSI layers 1-2. When a link between two telecommunication systems, or between a telecommunication system and a system simulator is required for testing purposes, the systems are normally connected directly to each other with the same transmission hardware that is utilized for physically connecting installed telecommunication systems in the field. Test equipment and test tools are then connected to the systems and the physical links between them for the purpose of monitoring the links and performing protocol analyses or other evaluation tests.

However, there are several disadvantages to these existing test methods. First, the test equipment and test tools utilized for development testing of telecommunication systems are very expensive. Second, these existing test methods still require the utilization of physical transmission hardware between the systems under test which increases the setup time and the cost of the testing process. Third, these methods require substantial access time to a valuable telecommunication system which may be in great

BNSDOCID: <WO__9735406A1_I_>

demand by a number of different development groups. Finally, the use of a physical telecommunication system and physical transmission hardware ties the tester to a physical location in close proximity to the system under test.

Although there are no known prior art teachings of a solution to the aforementioned deficiency and shortcoming, U.S. Patent Number 5,027,343 to Chan et al. (Chan) discusses subject matter that bears some relation to matters discussed herein. Chan discloses a test access system for remote testing of products in an integrated services digital network (ISDN) system. The protocols tested relate to OSI layers 1-3 which are concerned mainly with the establishment, holding, and release of a physical telecommunications path. Chan packetizes or encapsulates network messages, including layers 1-3, and utilizes a packet switching network to communicate test procedures from the tester to the system under test. The system under test de-encapsulates the packets, removes the network messages, and sends them to be processed.

Chan overcomes some of the disadvantages of utilizing the same physical transmission hardware for testing that is used for transmissions between telecommunication systems in the field. However, Chan is specifically designed to support remote testing of actual physical hardware. A local site which includes a tester is remotely linked to a system under test. Chan specifically states, however, that the patent is directed only to OSI layers 1-3, which are concerned with the physical transmission, routing, and switching of signals. Chan, therefore, is a hardware-dependent solution which is very sensitive to hardware timing requirements. Chan does not teach or suggest any method of connecting multiple telecommunication systems or system emulators, without physical transmission hardware, for the testing of OSI layers 3-7. Chan also does not teach or suggest a protocol simulator which utilizes protocol simulation software to simulate OSI layers 3-7 in a plurality of communications protocols. Nor does Chan teach or suggest connecting such a protocol simulator to a system under test without physical transmission hardware.

It would be a distinct advantage to have a system and method for linking a protocol simulator with multiple telecommunication systems and/or telecommunication systems and system emulators for development testing of software implementing OSI layers 3-7 in a telecommunications node that does not require the utilization of the same transmission hardware that is utilized for physically connecting installed

SUBSTITUTE SHEET (RULE 26)

telecommunication systems in the field. Such a system and method would eliminate the need for expensive test equipment and test tools for monitoring the communication links and performing protocol analyses or other evaluation tests. The system would be independent of the transmission hardware utilized for normal telecommunication transmissions, and would not be sensitive to hardware timing requirements. A protocol simulator and method of connecting the simulator to various networks is required for such a system to operate most effectively. The present invention provides such a protocol simulator and method.

SUMMARY OF THE INVENTION

In one aspect, the present invention is a protocol simulator for simulating open systems interconnection (OSI) layers 3-7 of a communications protocol and utilizing the OSI layers to test a target system connected to a network. The protocol simulator comprises protocol simulation software for simulating OSI layers 3-7 of the communications protocol, means for packaging the OSI layers into a protocol stack, means for generating test messages utilizing the protocol stack, means for simulating OSI layers 1 and 2, and means for transmitting the test messages over the network to the target system.

In another aspect, the present invention is a protocol simulator for simulating open systems interconnection (OSI) layers 3-7 of a plurality of communications protocols and utilizing the OSI layers to test selected ones of a plurality of target systems connected to a network. The protocol simulator comprises means for selecting one of the plurality of target systems to be tested and a plurality of protocol simulation software blocks, each of which simulates OSI layers 3-7 of a different communications protocol, each of the communications protocols being associated with one of the plurality of target systems. The protocol simulator also includes means for selecting the protocol simulation software block associated with the selected target system, means for packaging the OSI layers of the selected protocol simulation software block into a protocol stack, means for generating test messages utilizing the protocol stack, means for simulating OSI layers 1 and 2, and means for transmitting the test messages over the network to the selected target system.

BNSDOCID: <WO__9735406A1_I_>

In still another aspect, the present invention is a method of simulating open systems interconnection (OSI) layers 3-7 of a communications protocol and utilizing the OSI layers to test a target system connected to a network. The method comprises the steps of simulating OSI layers 3-7 of the communications protocol with protocol simulation software, packaging the OSI layers into a protocol stack, generating test messages utilizing the protocol stack, simulating OSI layers 1 and 2, and transmitting the test messages over the network to the target system.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be better understood and its numerous objects and advantages will become more apparent to those skilled in the art by reference to the following drawing, in conjunction with the accompanying specification, in which:

FIG. 1 (Prior Art) is an illustrative block diagram of an OSI stack illustrating the seven layers of the OSI model;

FIG. 2 is a simplified block diagram of a communication link utilizing an Internet socket between two OSI stacks connected at the layer 3 level in accordance with the teachings of the present invention;

FIG. 2A is a simplified block diagram illustrating the relationship between the various protocols utilized in the present invention;

FIG. 3 is a simplified block diagram of a communication link utilizing an Internet socket between two telecommunication system emulators utilizing protocols from the OSI model;

FIG. 4 is a simplified block diagram illustrating an embodiment of the present invention in which a protocol simulation tool has been modified to utilize an Internet socket and a LAN connection to link a call origination simulation with a call termination simulation;

FIG. 5 is a block diagram illustrating in more detail the protocol stacks and communication link of FIG. 3;

FIG. 6 is a block diagram of an embodiment of the present invention in which a protocol simulation test device interfaces with a telecommunication switch emulator via an Internet socket;

FIG. 7 is a simplified block diagram of an automated diagnostics system in which both a target telecommunication switch and a switch emulator are connected to a protocol simulator in accordance with the teachings of the present invention;

FIG. 8 is a more detailed block diagram of the emulator and PIG-tool of FIG. 7;

FIG. 9 is a simplified block diagram illustrating the simulation of a plurality of line interface cards (LICs) by the Protocol Simulator and their connection through an Internet socket and LAN to an emulator and a plurality of simulated signaling points;

FIG. 10 is an illustration of a computer-displayed Icon tool box which may be utilized by an operator to initiate testing with the development test system of the present invention;

FIG. 11 is an illustration of a computer-displayed shelf selection menu which enables an operator to select a real hardware system or an emulated system for testing purposes;

FIG. 12 is an illustration of a computer-displayed network map editor utilized to define the testing environment to the development test system;

FIG. 13 is an illustration of a computer-displayed network map illustrating a simulation in the protocol simulator of a simple location updating sequence performed in a Base Station Controller (BSC) in a Personal Communication System (PCS) mobile telecommunication system;

FIG. 14 is an illustration of a computer-displayed menu of emulated systems (virtual shelves) which the PIG-tool may access;

FIG. 15 is an illustration of a computer-display showing three status windows corresponding to the protocol simulator, the PIG-tool, and the emulator; and

FIG. 16 is a more detailed listing of the message monitor section of the PIG-tool status window of FIG. 15.

DETAILED DESCRIPTION OF EMBODIMENTS

During the development of new functions and services that are transmitted between telecommunication systems, OSI layers 1 and 2 (the physical hardware and the link to the network) are frequently unaffected. In contrast, layers 3-7 are almost always altered in some manner. In other words, the development of new functions and services does not always alter the means for transmitting information, but the information being

BNSDOCID: <WO__9735406A1_I_>

transmitted is almost always changed. Therefore, testing that involves only OSI layers 3-7 does not require the expensive and specialized transmission test equipment that is connected between network models since the physical layer is usually not affected. Instead, developers can take advantage of the fact that most telecommunication systems support some type of local area network (LAN) protocol.

FIG. 2 is a simplified block diagram of a communication link 20 utilizing an Internet socket 21 between two OSI stacks 22 and 23 connected at the layer 3 level. In accordance with the teachings of the present invention, the existing level 1 and 2 telecommunication protocols, the physical level and the data link level, are replaced by a LAN protocol such as, for example Ethernet. The telecommunication protocol is encapsulated by the Transmission Control Protocol/Internet Protocol (TCP/IP) and then transmitted via a LAN network utilizing an Internet socket interface 21. The Internet socket interface 21 may be a type of UNIX file that provides network communications between applications operating on different host processors. The socket interface allows application programs to communicate with each other. In general, an application program creates a TCP client socket, connects to a TCP server socket, and then sends or receives data across that interface.

The Internet socket interface 21 may be thought of as a generalization of the UNIX file access mechanism that provides an endpoint for communication. Application programs request the operating system to create a socket when one is needed. Sockets are created without binding them to specific destination addresses. The application may then supply a destination address each time it uses the socket (e.g., when sending datagrams), or it may choose to bind the destination address to the socket and avoid specifying the destination repeatedly (e.g., when making a TCP connection). A client socket connects to a server socket to allow communication between application programs.

A standard Internet protocol is the User Datagram Protocol (UDP). The UDP protocol includes a protocol port number allowing a sender to distinguish among multiple destinations (application programs) on a remote processor. UDP/IP sockets are utilized for establishing, maintaining, and removing communications between application programs while TCP/IP sockets are utilized to transfer data reliably.

FIG. 2A is a simplified block diagram illustrating the relationship between the various protocols utilized in the present invention. User processes 24 and 25 comprise OSI layers 5-7. Data is transferred between the user processes 24 and 25 utilizing the TCP protocol 26 (OSI layer 4). Other communications between the user processes utilize the UDP protocol 27 (OSI layer 4). Both data and communications then utilize the IP protocol 28 (OSI layer 3) and Ethernet 29 (OSI layers 1-2) to complete the connection between the user processes.

Internet sockets may be utilized with traditional operations such as "read" and "write". For example, once an application program creates a socket and creates a TCP connection from the socket to a destination address, the application program may use the "write" operation to send a stream of data across the connection. A receiving application program on the other end may use the "read" operation to receive the data.

FIG. 3 is a simplified block diagram of a communication link 30 utilizing an Internet socket 31 between two telecommunication system emulators 32 and 33 utilizing protocols from the OSI model. As utilized herein, the term "emulator" refers to a software program that emulates the hardware of a processing node and interprets the application software as if the application software was running on a target machine. The term "simulator" refers to a processor that is preprogrammed with responses to anticipated messages. In the preferred embodiment, a protocol simulator and a telecommunication system emulator run on processors that are UNIX-based. Therefore, these systems communicate with each other utilizing an Internet socket and a LAN-based network such as, for example Ethernet. Protocol based information is communicated from a transmitting system to a receiving system utilizing LAN protocols supported by the telecommunication systems. Communications between the protocol simulator and the telecommunication system emulator utilize the Internet socket to transmit packaged data from OSI layers 3-7 in TCP/IP format on the LAN-based network. The receiving emulator receives the information from the protocol simulator via a protocol interface gateway utilizing sockets, unpacks layers 3-7, and processes the information. A fixed OSI layer 2 in the telecommunications protocol under test (e.g. Message Transfer Part (MTP)) may also be utilized since the communication link between the protocol simulator and the telecommunication system emulator is controlled by a completely different protocol stack associated with the Internet socket.

BNSDOCID: <WO__9735406A1_I_>

When the communication link is between telecommunication nodes, OSI layers 3-7 may correspond to any ANSI Signaling System 7 (SS7), CCITT, or other compatible protocol stack. FIG. 3 illustrates the transport of a Transaction Capabilities Application Part (TCAP) protocol stack 33 to stack 32 or vice versa. Also illustrated are a Mobile Application Part (MAP) protocol stack 33a and an Integrated Services User Part (ISUP) protocol stack 33b. TCAP 34 and MAP 35 are level 7 applications while ISUP 36 is a level 4-7 application. The SS7 stack may be transported through the Internet socket 31 over a LAN using Ethernet to replace the MTP layers 1 and 2. The physical target telecommunications node or telecommunication system emulator accepts the information on a TCP/IP link on the LAN and decodes the layer 3-7 information. Some telecommunication nodes may require modification for communication over TCP/IP with a LAN connection to replace their normal MTP layers 1 and 2. Such modifications are well known in the art and will not be expanded upon herein.

This communication process may also be utilized to communicate between a UNIX processor that outputs telecommunication signaling and a physical target telecommunication node that is equipped to receive the LAN information and extract the incoming signaling information.

FIG. 4 is a simplified block diagram illustrating an embodiment of the present invention in which a protocol simulation tool 41 has been modified to utilize Internet sockets 42 and 43, and a LAN connection 44 to link a call origination simulation 45 with a call termination simulation 46. In this embodiment, telecommunication protocol simulation software for layers 3-7 of the OSI model is transmitted from the call origination simulation 45 to the call termination simulation 46, and vice versa, without requiring any external network or hardware. The call origination simulation 45 comprises test script software that performs the call origination functions. These scripts are then built into layers 3-7 at block 47 and sent to a UNIX adaptor 48. The UNIX adaptor 48 packages the layers in TCP/IP format at block 49 for transmission through the Internet socket 42, the LAN connection 44, and the Internet socket 43 to a receiving side 50 of the UNIX adaptor 48 where the layers are unpacked. The layers are then unbuilt at block 51, and are sent to the call termination simulation 46 in the simulation tool 41. In this manner, signals from the simulation tool's software package are looped back into the simulation tool itself to enable test script verification of OSI layers 3-7.

BNSDOCID: <WO 9735406A1 >

Processing continues in a dialog as determined by the pre-programmed test script, with each script designed to send and receive protocol messages built with layers 3-7 of the communications protocol being tested.

FIG. 5 is a block diagram illustrating in more detail the protocol stacks 22 and 23, and the communication link 20 of FIG. 2. Protocol stack 22 may be a UNIX-based test application which communicates with a unit under test 52 via the Internet socket 21. The test application includes a MTP comprising the OSI model protocol stack. At the MTP User level (layer 7), there is a UNIX-based application 53 utilizing an MTP protocol for communications with the MTP User. Moving down the protocol stack, there is a Signaling Connection Control Part (SCCP) layer 54 and a MTP layer 3 55 which performs signaling network functions. Below MTP layer 3 are MTP layers 1 and 2 where, in the present invention, the normal MTP layers 1 and 2 hardware related components have been replaced by a UNIX application 56.

The Internet socket 21 transports packaged data from the UNIX application 56 in the UNIX-based test application to a second UNIX application 57 simulating the hardware components of normal MTP layers 1 and 2 in the unit under test 52. Specifications made between the UNIX applications 56 and 57 allow them to communicate the OSI layer 3-7 information between the applications without dynamic assignment of layer 2 information by using UNIX utilities, with or without utilizing LAN communication and protocols.

When utilized in the specific application of communicating the ANSI SS7 protocol, SS7 messages are packaged from OSI layer 7 to the layer 3 level and passed from the test application 22 to the unit under test 52 using the UNIX platform and utilities. The unit under test receives the data for processing, executes functions in the unit under test, and may package response information for OSI layers 3-7 for handling in the test application.

FIG. 6 is a block diagram of an embodiment of the present invention in which a protocol simulation test device (protocol simulator) 60 interfaces via Internet sockets 61 and 62 with a UNIX process 63 running a target system emulation. The protocol simulator 60 comprises a UNIX process running standard simulation tools. The standard tools include test script software 64 that performs the call origination functions. These scripts are then built into OSI layers 3-7 in block 65 and sent to a UNIX adaptor 66. The

BNSDOCID: <WO 9735406A1 I_>

UNIX adaptor packages the layers in TCP/IP format for transmission through Internet sockets 61 and 62 and a LAN connection 67 to the UNIX process running the target system emulation 63. The emulation process 63 includes a protocol interface gateway (PIG-tool) 68 and a target system emulator 69. The PIG-tool 68 unpacks the OSI layers received from the socket in TCP/IP format and converts them to processor instructions. The instructions are then passed to the target system emulator 69.

The target system emulator 69 emulates the hardware of a target telecommunication node operating in the UNIX environment, and is capable of sending and receiving signaling information at the OSI layer 3 level or higher. The emulator 69 may communicate with an interfacing application using Internet sockets or other UNIX facilities such as pipes or interface 67 to send and receive the OSI layer 3-7 information. After performing required functions on the received information, the emulator may respond with communications back to the protocol simulator 60 using the same method of packaging in TCP/IP format, and transmission through Internet sockets to transfer the data to the protocol simulator for verification by the test script.

FIG. 7 is a simplified block diagram of an automated diagnostics system 70 in which both a target telecommunication switch 71 and a switch emulator 72 are connected to a protocol simulator 73 in accordance with the teachings of the present invention. The protocol simulator 73 may be alternately referred to in various computer-generated figures as a Protocol Adaptable State Machine (PASM) or a Message Generator Traffic Simulator (MGTS). The protocol simulator 73 enables a user to select either an actual target hardware node or a node emulated by software for the testing of OSI layers 3-7. If a hardware node is selected (for example, the target telecommunication switch 71), a communications manager 74 identifies this selection to a protocol simulation socket adaptation module (protocol simulation adaptor) 75 which then selects the proper protocol simulation software 76 and protocol stack 77 for the hardware node. Messages in UNIX-based TCP/IP protocol are sent through one of a plurality of Internet sockets 78 to a LAN 79. The messages exit the LAN through an Internet socket 81 and are directed to a UNIX adaptor 82. The UNIX adaptor 82 translates the TCP/IP protocol to SS7 messages which are understood by the target telecommunication switch 71.

BNSDOCID: <WO_9735406A1_>

If an emulator is selected (for example the switch emulator 72), the communications manager 74 identifies this selection to a protocol simulation adaptor 75 which then selects the proper protocol simulation software 76 and protocol stack 77 for the emulated system. Messages in UNIX-based TCP/IP protocol are sent through one of the plurality of Internet sockets 78 to the LAN 79. The messages exit the LAN through a gateway Internet socket 83 and are directed in the preferred embodiment to a protocol interface gateway (PIG-tool) 84.

The protocol simulation adaptor 75 is an simulation of a Message Generation Traffic Simulator (MGTS) hardware chassis running as a process or a group of processes on a workstation. The protocol simulation adaptor 75 includes protocol simulation software 76 and associated protocol stacks 77. It offers the ability to perform test sequence verification without accessing a hardware chassis. Users may debug and trouble shoot test sequences prior to connecting the target hardware node 71 or 72.

For communication with an emulator, the protocol simulation adaptor 75 utilizes the SS7 TCP/IP protocol. Test messages may also include header octets that are proprietary between the protocol simulator and target or emulated telecommunication node to identify the source of the message, message length, protocol revision, etc. The test messages are communicated through one of the Internet sockets 78 over the LAN 79 such as, for example Ethernet, through the gateway Internet socket 83 to the PIG-tool 84. The PIG-tool 84 strips off the TCP/IP protocol and headers, and converts the test messages to CPU instructions. The CPU instructions are then sent to the emulator 72 for processing and exercising the software to be tested. The emulator 72 includes the application software blocks 85 from the simulated telecommunication node as well as software modules that emulate the hardware of the target node. The emulator exercises the software to be tested, verifies the use of the simulated communications protocol with the target telecommunication node, and responds back toward the protocol simulator 73. Therefore, after a user develops test sequences for an application on the emulator 72, the protocol simulator 73 may be utilized to test the application on the target hardware 71 using the same test sequences.

The protocol simulator 73 may also test applications through the Internet socket 78 and through a physical interface 86 with the target host hardware 71 simultaneously.

BNSDOCID: <WO 9735406A1_I >

Additionally, the protocol simulator 73 may control other testing devices (not shown) through the Internet socket 78 and the LAN 79.

FIG. 8 is a more detailed block diagram of the UNIX-based emulator 72 and PIG-tool 84 of FIG. 7. The protocol simulator 73, which includes hooks into the protocol simulation adaptor 75, runs the protocol simulation software 76. In the preferred embodiment, the protocol simulator 73 is configured to service 12 users, and utilizes a Sparc 20 Server from SUN Microsystems having 32 Mb RAM and 1,000 Mb swap space on the server's hard drive. The UNIX-based emulator 72 and PIG-tool 84 may be operated on a SUN Sparc 5 Workstation having 32 Mb RAM and 200 Mb swap space on the workstation's hard drive. For an embodiment servicing a single user, the protocol simulator 73 may also be operated on a SUN Sparc 5 Workstation. These hardware configurations are described for illustrative purposes only, and are not intended to limit the scope of the present invention which may be implemented with other hardware configurations.

The PIG-tool 84 receives raw SS7 octets (OSI layers 3-7) 87, with or without proprietary header octets, from the gateway Internet socket 83 which is connected to the Ethernet LAN 79. The SS7 octets enter the PIG-tool 84 through a Virtual Patch Panel (VPP) 88. Within the PIG-tool, the SS7 octets are collected until a complete SS7 Message Signal Unit (MSU) is received.

The PIG-tool 84 determines the source of the MSU through the Internet socket on which the MSU was received. If the MSU was received from the protocol simulator 73, the proprietary header octets, if present, are stripped off and processed. If the MSU was received from the emulator 72 in the form of one or more emulator instructions, these instructions are processed by the PIG-tool into a MSU buffer (not shown).

When the MSU has been fully received, the PIG-tool 84 consults routing tables which are specified by the user and are internally stored and maintained in the VPP 88. The routing tables associate source entities with destination entities. Once the destination of the MSU is determined from the routing tables, the PIG-tool processes the message into the suitable format for the destination. If the destination is the protocol simulator 73, this processing involves the optional encapsulation of the MSU with the proprietary header, if one was utilized. If the destination of the MSU is the emulator 72,

the MSU is broken up into the appropriate number of emulator instructions and sent into the emulator for processing by the software instruction being tested.

FIG. 9 is a simplified block diagram illustrating the simulation of a plurality of line interface cards (LICs) 91 by the protocol simulator 73 and their connection through Internet sockets 78 and 83 and LAN 79 to an emulator 92 and a plurality of simulated signaling points 93. In one embodiment of the present invention, the protocol simulator 73 may simulate up to 16 LICs 91, and the emulator 92 may connect to the LAN with up to 16 signaling points 93, all of which are connected through Internet sockets 78 and 83 and an Ethernet LAN 79. This number of LICs and signaling points is not a limitation on the present invention, but is presented as an exemplary embodiment only. In other embodiments, a greater or lesser number of LICs and signaling points may be utilized.

Detailed Example

FIG. 10 is an illustration of a computer-displayed Icon tool box 100 which may be utilized by an operator to initiate testing with the development test system of the present invention. The operator may control the system from either the emulator workstation or the protocol simulator workstation. A first PIG-tool Icon 101 is labeled "socket", and a second PIG-tool Icon 102 is labeled "normal". The PIG-tool 84 is capable of interfacing with two different emulators, one that is socket-based, and one "normal" emulator that is not socket based, and these Icons are utilized to select the type of emulator to interface with the PIG-tool. A "MGTS" Icon 103 is utilized to initiate the protocol simulation software 76 in the protocol simulator 73.

FIG. 11 is an illustration of a computer-displayed shelf selection menu 110 which enables an operator to select a real hardware system or an emulated system for testing purposes. Selection of a menu item labeled "real" connects the protocol simulator 73 to a real target hardware system. Selection of a menu item beginning with "exu..." connects the protocol simulator 73 to an emulated system. The emulated systems are divided into three categories - small, medium, and large - depending on the number of line interface cards (LICs) 91 associated with the selected menu item.

FIG. 12 is an illustration of a computer-displayed network map editor 120 utilized to define the testing environment to the development test system of the present invention. A set of simulated nodes 121-130 surrounds a node under test (test node) 131

BNSDOCID: <WO__9735406A1_I_>

positioned in the center of the display. The test node 131 may be a real target hardware system or an emulated system, depending on the type of system selected by the operator utilizing the shelf selection menu 110 of FIG. 11. If the operator chooses a real target hardware system, then the connections 132 between the test node 131 and the surrounding nodes 121-130 are physical links. If the operator chooses an emulated system, then the connections 132 between the test node 131 and the surrounding nodes 121-130 are simulated SS7 links. Each node is labeled with a signaling point code (e.g., 7-9-60) and a short name for the node being simulated (e.g., MSC 4).

FIG. 13 is an illustration of a computer-displayed network map (simulation script) illustrating a simulation in the protocol simulator of a simple location updating sequence performed in a Base Station Controller (BSC) in a Personal Communication System (PCS) mobile telecommunication system. When a mobile subscriber turns on a mobile telephone, the location updating sequence is initiated as shown at step 136 of the protocol simulation of the BSC. The BSC then sends a location updating request message 137 to its Mobile Switching Center (MSC) which, in turn, updates the location of the mobile subscriber in the subscriber's home location register (HLR). A confirmation message is then returned from the MSC to the BSC at 138.

In FIG. 13, simulated nodes that receive messages are illustrated with arrows to the left of a center vertical line. Simulated nodes that transmit messages are illustrated with arrows to the right of the center vertical line. Simulated nodes that both receive and transmit messages are illustrated with arrows on both sides of the center vertical line. A simulated node 139 labeled "Loop" is positioned in the center of the display and is a holding point where the simulation waits for another message to be received. A timer 141 monitors the loop for activity, and if there is no activity for a preset time period (e.g., 10 seconds), the test sequence is stopped.

The process starts at node 136 and moves to node 137 where the BSC transmits the location updating request message to the MSC. In the development test system of the present invention, this request message may go to a real target MSC and HLR, or may go through the PIG-tool 84 to an emulator with traffic handling software and a built-in HLR. The location updating request message is in SS7 format, and the PIG-tool converts it to emulator instructions. When the emulator processes and responds to the request, the PIG-tool converts the response back to SS7 format for transmission to the

BSC simulation. When the MSC emulator responds with the request message, the BSC simulation receives a connection confirmation (CC) message at node 138. The process then moves to the loop node 139 and waits for the MSC emulator to respond to the request message. If the location updating request is accepted, the BSC simulation receives a location updating accept message at node 142. The link to the MSC/emulator is then cleared at node 143, and a release complete message is sent at node 144. The process then ends at 145.

FIG. 14 is an illustration of a computer-displayed menu 146 of protocol simulation systems (virtual shelves) 148 which the PIG-tool 84 may access. This list is identical to the protocol simulation systems listed in the shelf selection menu 110 in the protocol simulator illustrated in FIG. 11. When a protocol simulation system is selected (e.g., exuboge_large 149), a list of available nodes or line interface cards (LICs) 151 associated with the selected system is displayed. The list of LICs 151 also includes a reference to signaling points (STs) which correspond to data in the emulated telecommunication system. Once the operator selects the protocol simulation system and LICs that are to be utilized in testing, clicking on the button 152 labeled "MGTS Virtual Shelf" establishes a connection to the gateway Internet socket 83 (FIG. 7). When that connection is made, clicking on the "Emulator" button 153 establishes a connection to the emulator. Thus, the PIG-tool 84 makes a connection between a simulated LIC generated by the protocol simulator and a signaling point in the emulated system.

FIG. 15 is an illustration of a computer-display 155 showing three status windows corresponding to the protocol simulator (labeled as PASM Simulation) 156, the PIG-tool 157, and the emulated system (labeled as EmuTool) 158. An upper section 159 of the protocol adaptable state machine (PASM) Simulation window 156 indicates that the location updating sequence has been performed in the Mobile Switching Center/Home Location Register (MSC/HLR) and that the BSC protocol simulation has been verified by the protocol simulator. A middle section 161 is a log showing the times that particular messages were transmitted and received. A bottom section 162 illustrates the simulation script 135 of FIG. 13 with the path of the simulation indicated by arrows.

The PIG-tool window 157 includes an upper section 163 that shows the menu of protocol simulation systems (virtual shelves) 148 which the PIG-tool 84 may access. A lower section 164 is a message monitor which provides data about the messages being

BNSDOCID: <WO_ 9735406A1 I >

transmitted and received between the protocol simulator 73 and the PIG-tool 84, as the data passes through the gateway socket 83 (FIG. 7).

The emulator window 158 provides information on commands and data going from the PIG-tool 84 to the emulator 72. As shown in FIG. 9, a converter 94 in the PIG-tool 84 converts the SS7 hex to emulator code and sends it to an application programming interface (API) 95. An upper emulator log section 165 (FIG. 15) shows the commands received in emulator code 166 and the signal connection point 167 for each received command. A lower command section 168 (partially concealed) provides a condensed version of the commands sent from the PIG-tool 84 to the emulator 72.

FIG. 16 is a more detailed listing of the message monitor section 164 of the PIG-tool status window 157 of FIG. 15. Data is provided on all the LICs that have been selected for testing. The window 164 displays a source for each message (e.g., "Received from MGTS Platform") 171, proprietary header information 172, and the data carried in the message 173. The header information 172 indicates the revision, protocol, direction, and the length of the SS7 message. The message data 173 is presented in octets (8 bits) of SS7 data in hexadecimal representation. The PIG-tool 84 may send messages to the emulator 72 at the same time that it is receiving messages from the protocol simulator 73 (FIG. 7).

It is thus believed that the operation and construction of the present invention will be apparent from the foregoing description. While the method, apparatus and system shown and described has been characterized as being preferred, it will be readily apparent that various changes and modifications could be made therein without departing from the spirit and scope of the invention as defined in the following claims.

WHAT IS CLAIMED IS:

1. A protocol simulator for simulating open systems interconnection (OSI) layers 3-7 of a communications protocol and utilizing said OSI layers to test a target system connected to a network, said protocol simulator comprising:

protocol simulation software for simulating OSI layers 3-7 of said communications protocol;

means for packaging the OSI layers into a protocol stack;

means for generating test messages utilizing said protocol stack;

means for simulating OSI layers 1 and 2; and

means for transmitting said test messages over said network to said target system.

2. The protocol simulator of claim 1 wherein said network is a local area network (LAN).

3. The protocol simulator of claim 2 wherein said LAN is an Ethernet LAN.

4. The protocol simulator of claim 1 wherein said network is a wide area network (WAN).

5. The protocol simulator of claim 2 wherein said means for simulating OSI layers 1 and 2 is a UNIX application.

6. The protocol simulator of claim 5 wherein said means for transmitting said test messages over said network to said target system includes a UNIX socket.

7. The protocol simulator of claim 6 wherein said means for packaging the OSI layers into a protocol stack includes a UNIX adaptor for packaging the layers in Transmission Control Protocol/Internet Program (TCP/IP) format for transmission through the UNIX socket.

8. A protocol simulator for simulating open systems interconnection (OSI) layers 3-7 of a plurality of communications protocols and utilizing said OSI layers to test selected ones of a plurality of target systems connected to a network, said protocol simulator comprising:

means for selecting one of said plurality of target systems to be tested;

a plurality of protocol simulation software blocks, each of which simulates OSI layers 3-7 of a different communications protocol, each of said communications protocols being associated with one of said plurality of target systems;

BNSDOCID: <WO__ 9735406A1 I >

means for selecting the protocol simulation software block associated with the selected target system;

means for packaging the OSI layers of the selected protocol simulation software block into a protocol stack;

means for generating test messages utilizing said protocol stack;

means for simulating OSI layers 1 and 2; and

means for transmitting said test messages over said network to said selected target system.

9. The protocol simulator of claim 8 wherein said network is a local area network (LAN).

10. The protocol simulator of claim 9 wherein said LAN is an Ethernet LAN.

11. The protocol simulator of claim 8 wherein said network is a wide area network (WAN).

12. The protocol simulator of claim 9 wherein said means for selecting one of said plurality of target systems to be tested includes means for selecting a target hardware node or an emulated target node as said target system.

13. The protocol simulator of claim 9 wherein said means for simulating OSI layers 1 and 2 is a UNIX application.

14. The protocol simulator of claim 13 wherein said means for transmitting said test messages over said network to said selected target system includes a UNIX socket.

15. The protocol simulator of claim 14 wherein said means for packaging the OSI layers into a protocol stack includes a UNIX adaptor for packaging the layers in Transmission Control Protocol/Internet Program (TCP/IP) format for transmission through the UNIX socket.

16. A method of simulating open systems interconnection (OSI) layers 3-7 of a communications protocol and utilizing said OSI layers to test a target system connected to a network, said method comprising the steps of:

simulating OSI layers 3-7 of said communications protocol with protocol simulation software;

packaging the OSI layers into a protocol stack;

generating test messages utilizing said protocol stack;

simulating OSI layers 1 and 2; and

transmitting said test messages over said network to said target system.

17. The method of claim 16 wherein said step of transmitting said test messages over said network includes transmitting said test messages over a local area network (LAN).

18. The method of claim 17 wherein said step of transmitting said test messages over a LAN includes transmitting said test messages over an Ethernet LAN.

19. The method of claim 16 wherein said step of transmitting said test messages over said network includes transmitting said test messages over a wide area network (WAN).

20. The method of claim 17 wherein said step of simulating OSI layers 1 and 2 includes simulating OSI layers 1 and 2 with a UNIX application.

21. The method of claim 20 wherein said step of transmitting said test messages over said network to said target system includes transmitting said test messages with a UNIX socket.

22. The method of claim 21 wherein said step of packaging the OSI layers into a protocol stack includes packaging the layers in Transmission Control Protocol/Internet Program (TCP/IP) format for transmission through the UNIX socket.

23. A method of simulating open systems interconnection (OSI) layers 3-7 of a plurality of communications protocols and utilizing said OSI layers to test selected ones of a plurality of target systems connected to a network, said method comprising the steps of:

selecting one of said plurality of target systems to be tested;

simulating with a plurality of protocol simulation software blocks, OSI layers 3-7 of each of the plurality of communications protocols, each of said communications protocols being associated with one of said plurality of target systems;

selecting the protocol simulation software block associated with the selected target system;

packaging the OSI layers of the selected protocol simulation software block into a protocol stack;

generating test messages utilizing said protocol stack;

simulating OSI layers 1 and 2; and

BNSDOCID: <WO__9735406A1_I_>

transmitting said test messages over said network to said selected target system.

24. The method of claim 23 wherein said step of transmitting said test messages over said network includes transmitting said test messages over a local area network (LAN).

25. The method of claim 24 wherein said step of transmitting said test messages over a LAN includes transmitting said test messages over an Ethernet LAN.

26. The method of claim 25 wherein said step of transmitting said test messages over said network includes transmitting said test messages over a wide area network (WAN)..

27. The method of claim 24 wherein said step of selecting one of said plurality of target systems to be tested includes selecting a target hardware node or an emulated target node as said target system.

28. The method of claim 27 wherein said step of simulating OSI layers 1 and 2 includes simulating OSI layers 1 and 2 with a UNIX application.

29. The method of claim 28 wherein said step of transmitting said test messages over said network to said selected target system includes transmitting said test messages with a UNIX socket.

30. The method of claim 29 wherein said step of packaging the OSI layers into a protocol stack includes packaging the layers in Transmission Control Protocol/Internet Program (TCP/IP) format for transmission through the UNIX socket.

FIG.1
PRIOR ART

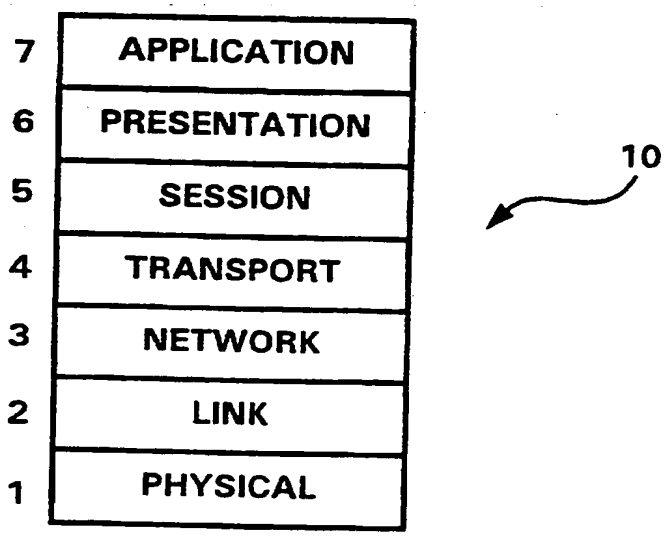
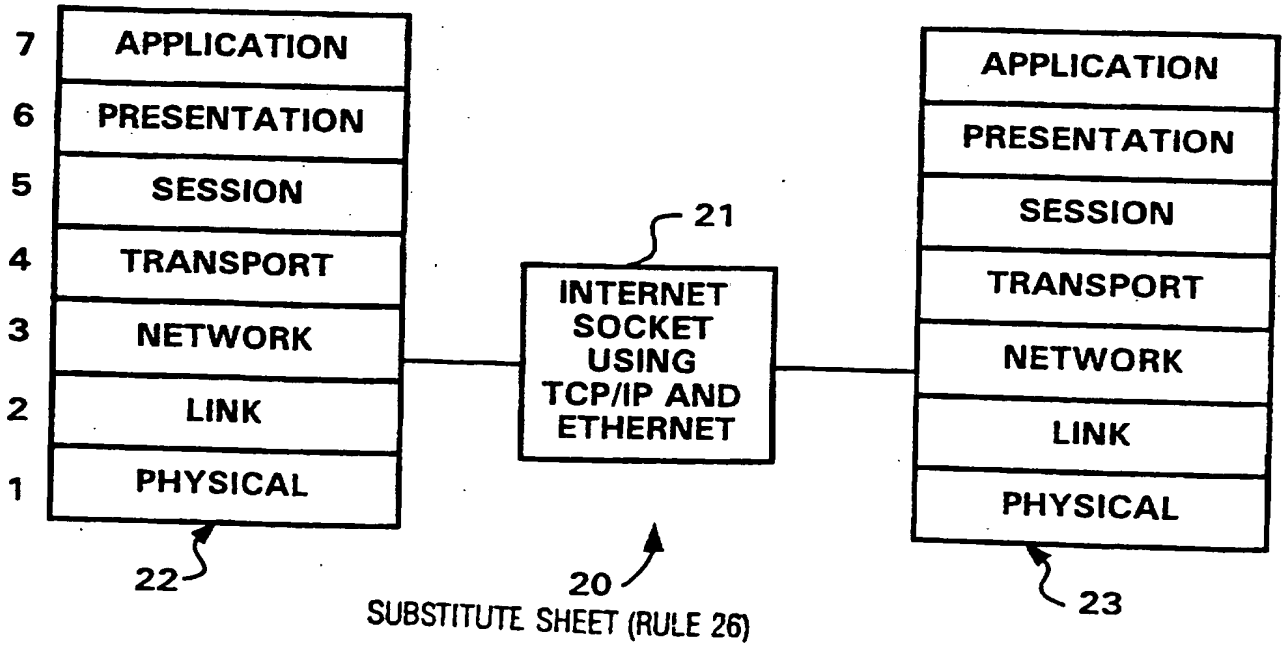
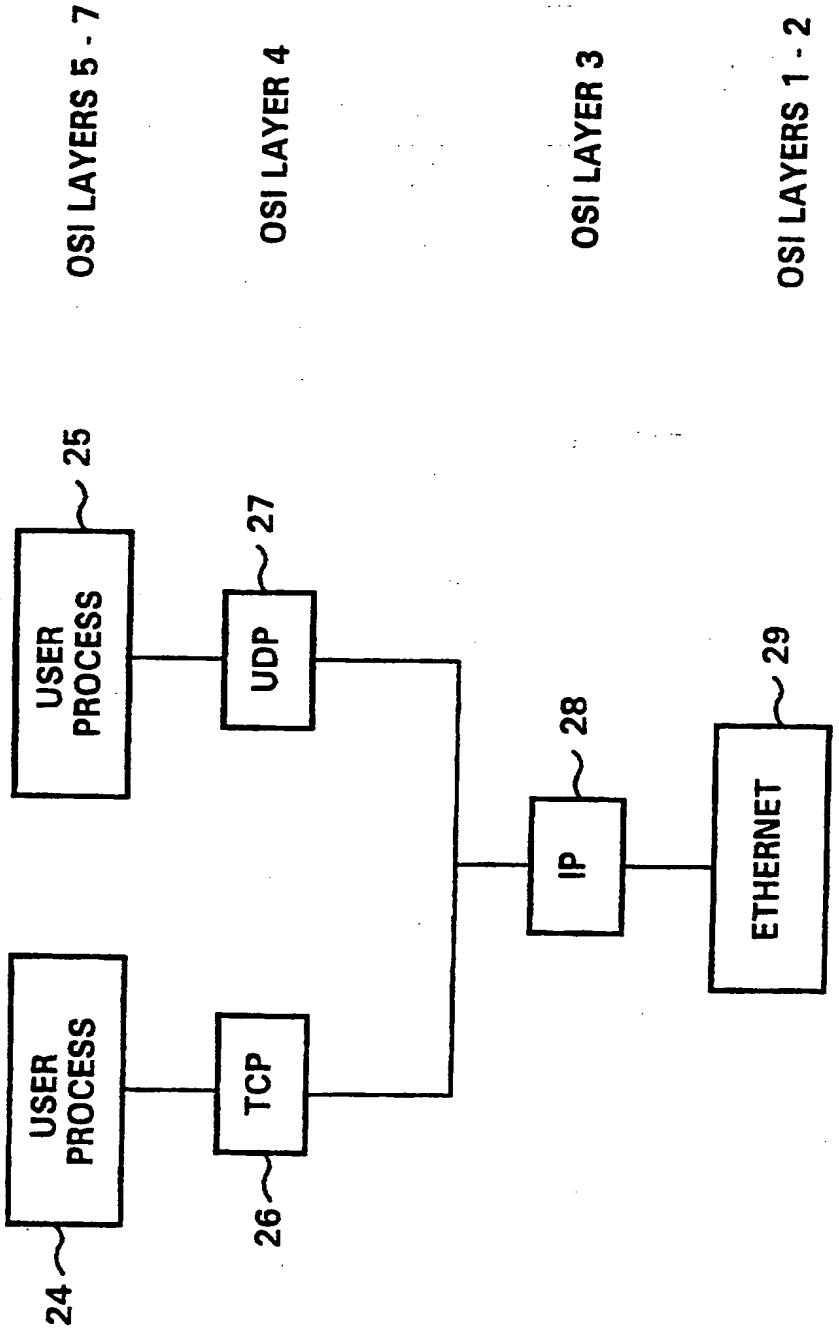


FIG.2



BNSDOCID: <WO__ 9735406A1.I.>

FIG.2A


OSI LAYERS 5 - 7
OSI LAYER 4
OSI LAYER 3
OSI LAYERS 1 - 2
USER PROCESS
25
UDP
27
USER PROCESS
24
TCP
26
IP
28
ETHERNET
29

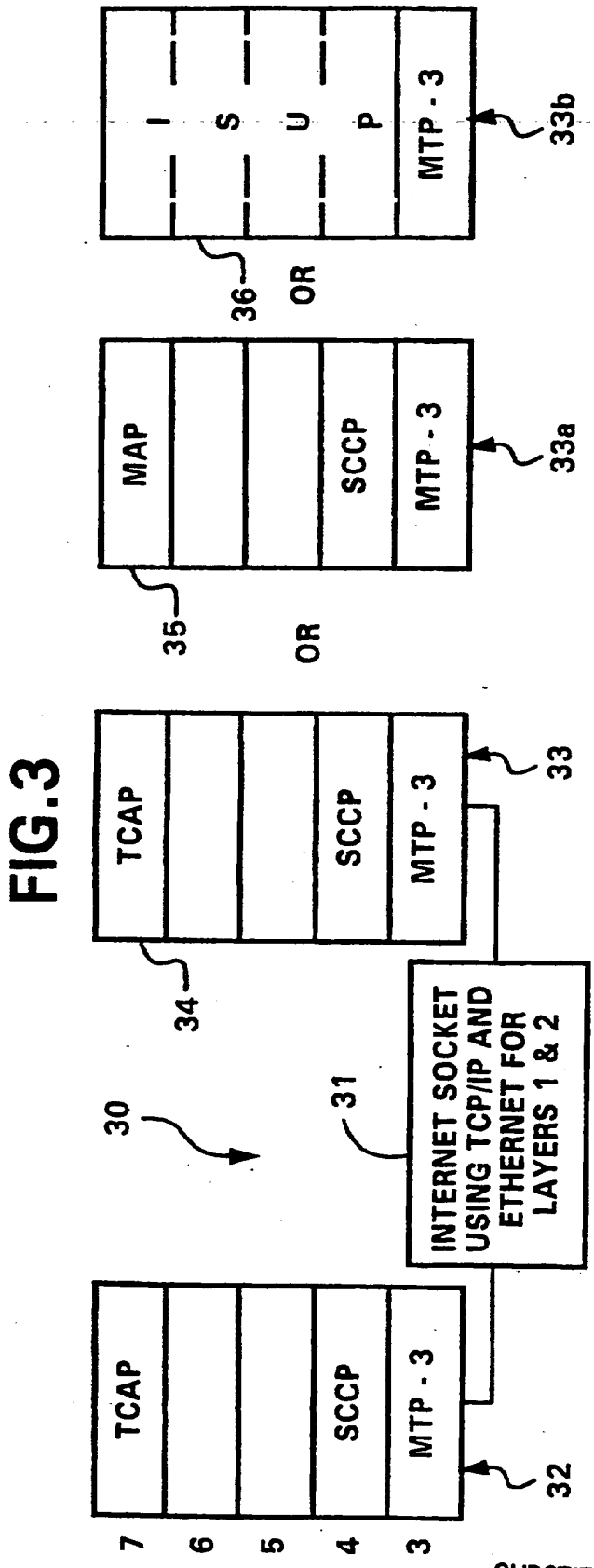


FIG.3

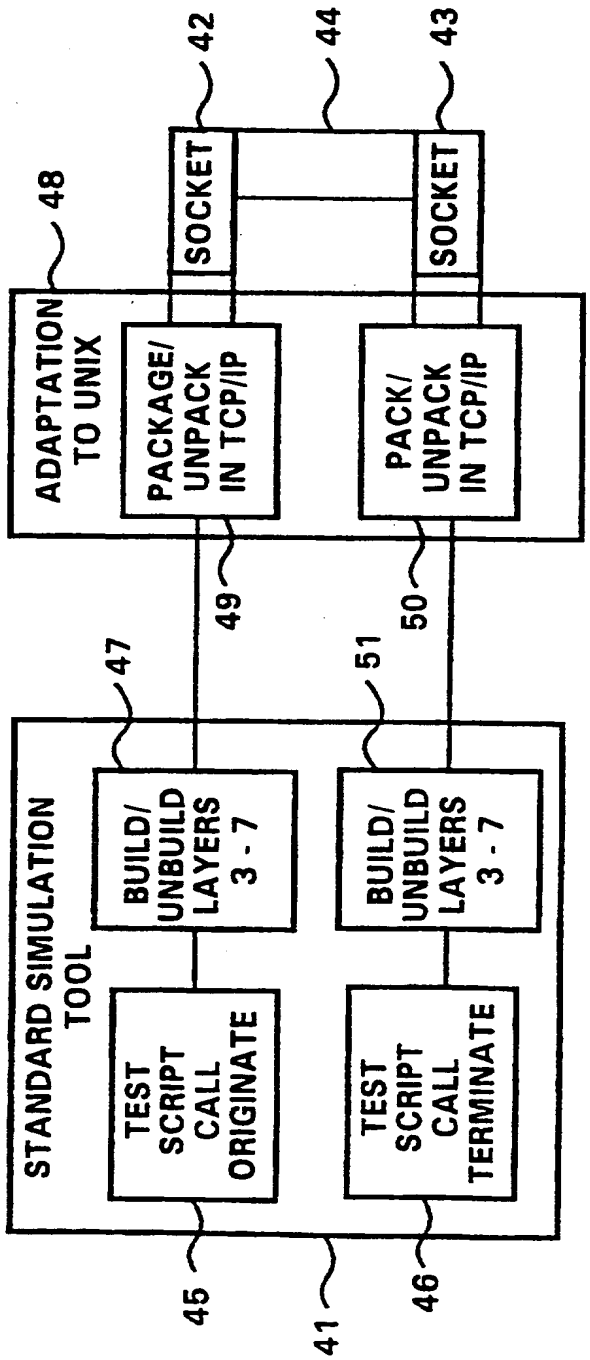
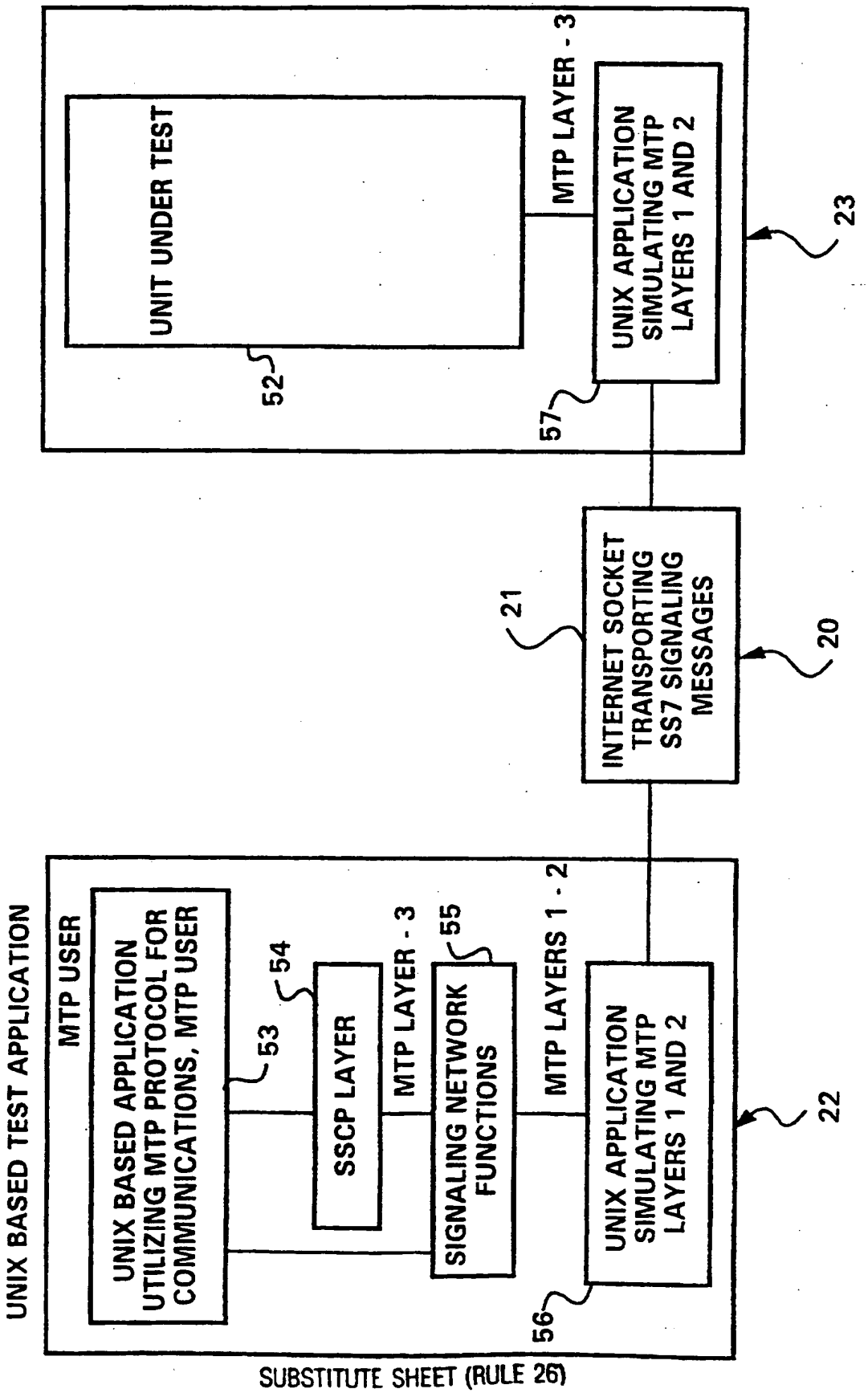


FIG.4

BNSDOCID: <WO__9735406A1_I_>

FIG.5


UNIT UNDER TEST
52
MTP LAYER - 3
UNIX APPLICATION SIMULATING MTP LAYERS 1 AND 2
57
23
21
INTERNET SOCKET TRANSPORTING SS7 SIGNALING MESSAGES
20
UNIX BASED TEST APPLICATION
MTP USER
UNIX BASED APPLICATION UTILIZING MTP PROTOCOL FOR COMMUNICATIONS, MTP USER
53
54
SSCP LAYER
MTP LAYER - 3
55
SIGNALING NETWORK FUNCTIONS
MTP LAYERS 1 - 2
UNIX APPLICATION SIMULATING MTP LAYERS 1 AND 2
56
22

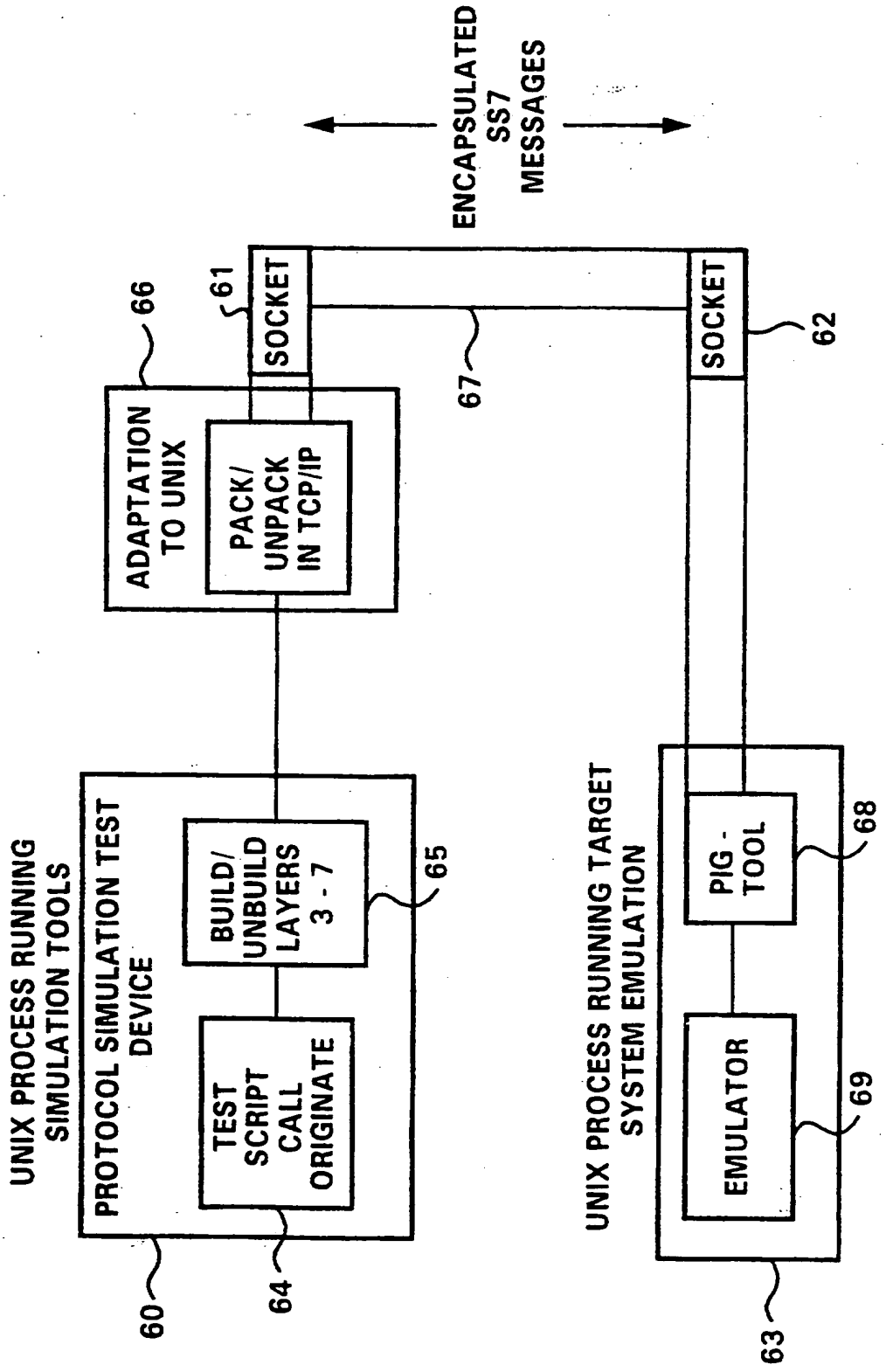


FIG.6

BNSDOCID: <WO__9735406A1 I >

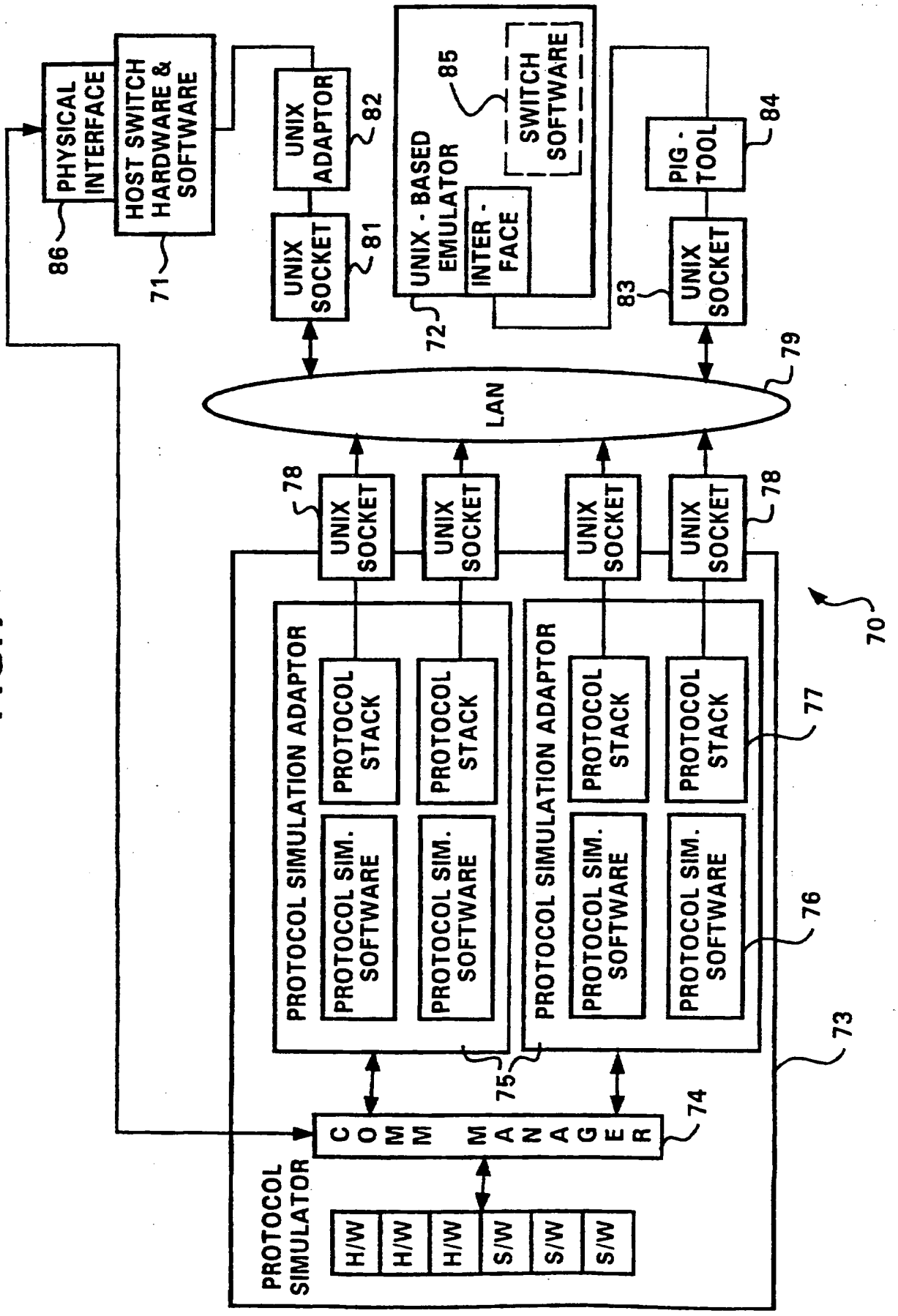
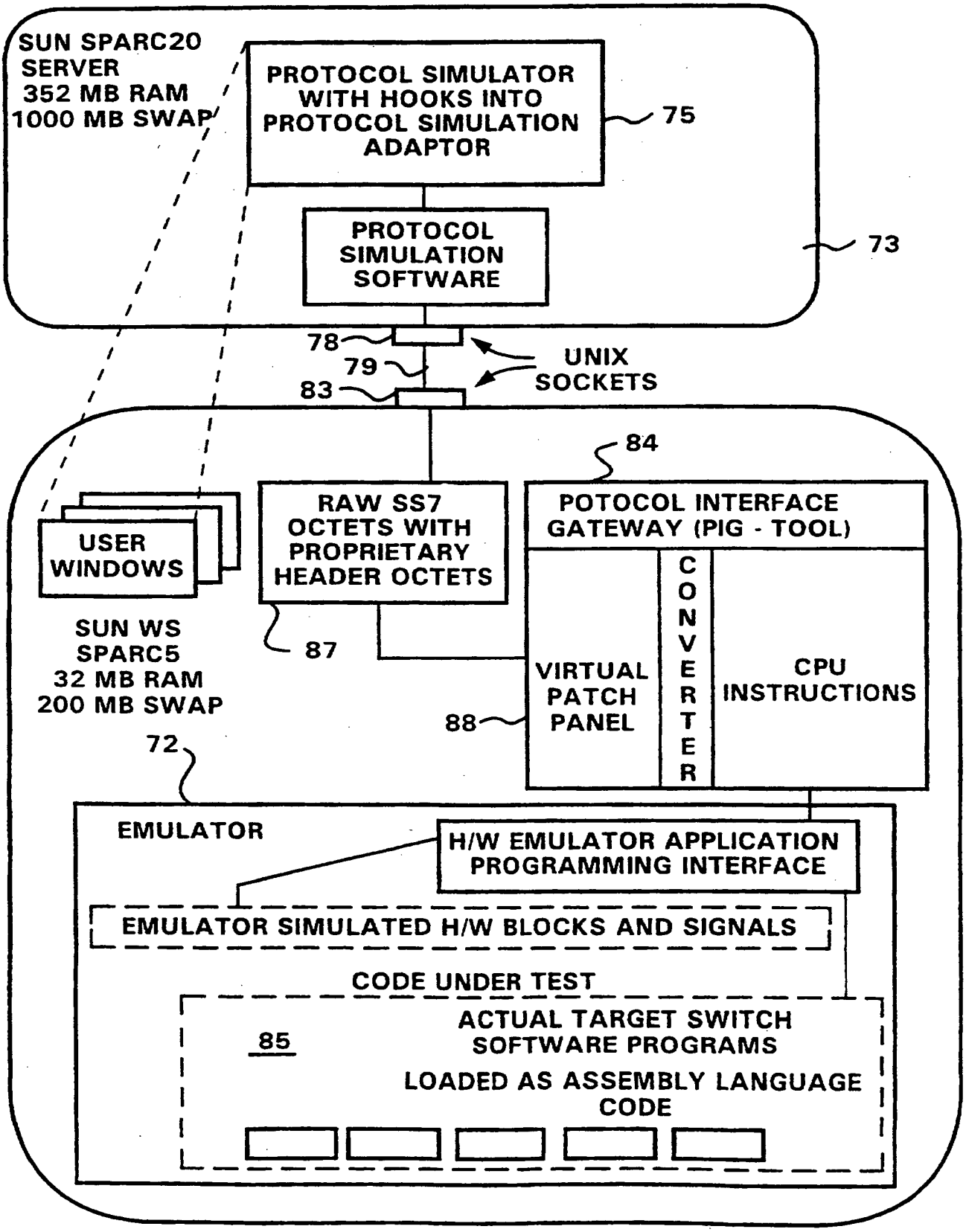

PHYSICAL INTERFACE
HOST SWITCH HARDWARE & SOFTWARE
UNIX ADAPTOR
UNIX - BASED EMULATOR
SWITCH SOFTWARE
INTER - FACE
PIG - TOOL
UNIX SOCKET
LAN
PROTOCOL SIMULATION ADAPTOR
PROTOCOL STACK
PROTOCOL SIM. SOFTWARE
COMMANAGER
PROTOCOL SIMULATOR
H/W
S/W
86
71
82
85
84
81
72
83
79
78
77
76
75
74
73
70

FIG.7

FIG.8



BNSDOCID: <WO__9735406A1_I_>

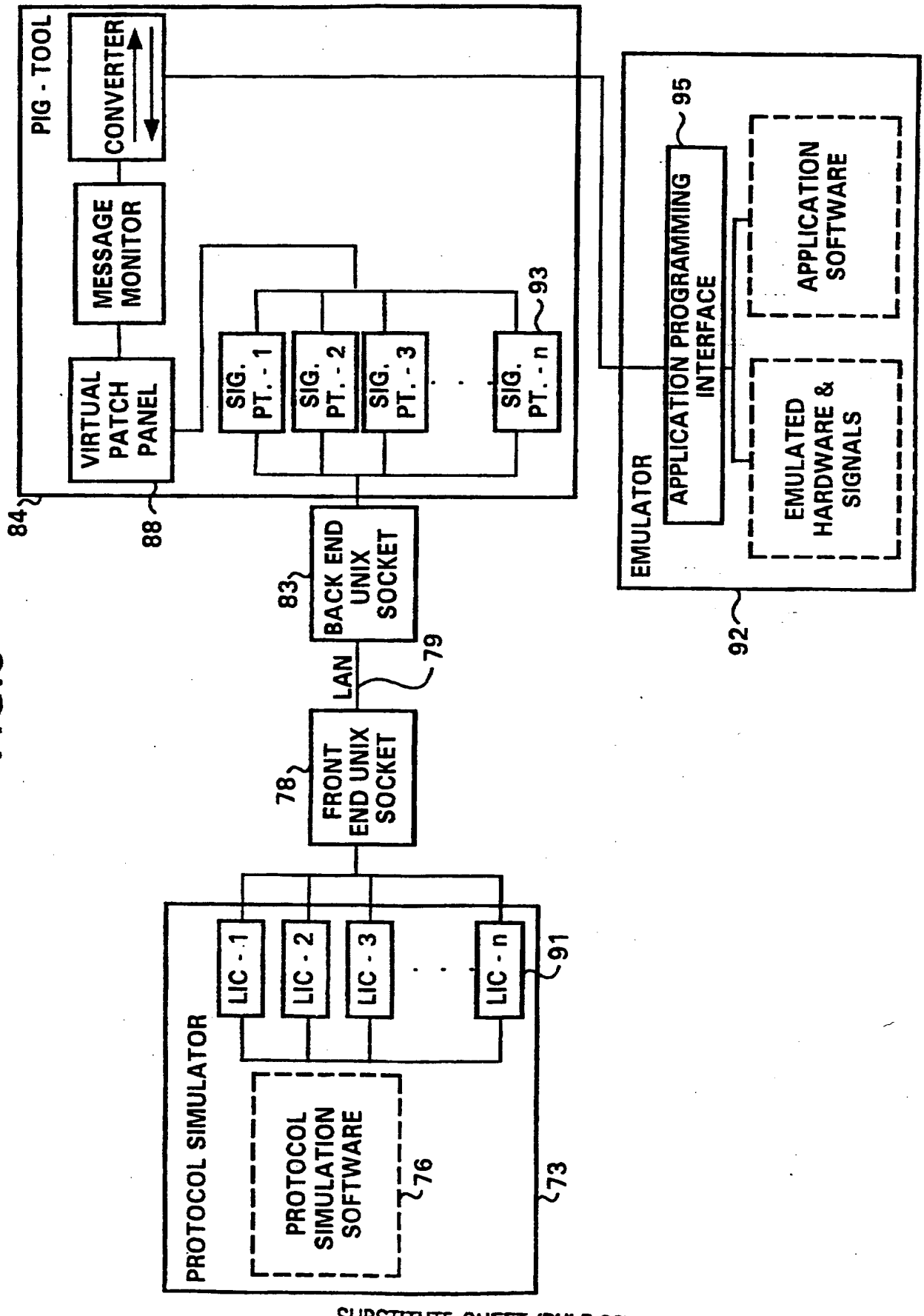
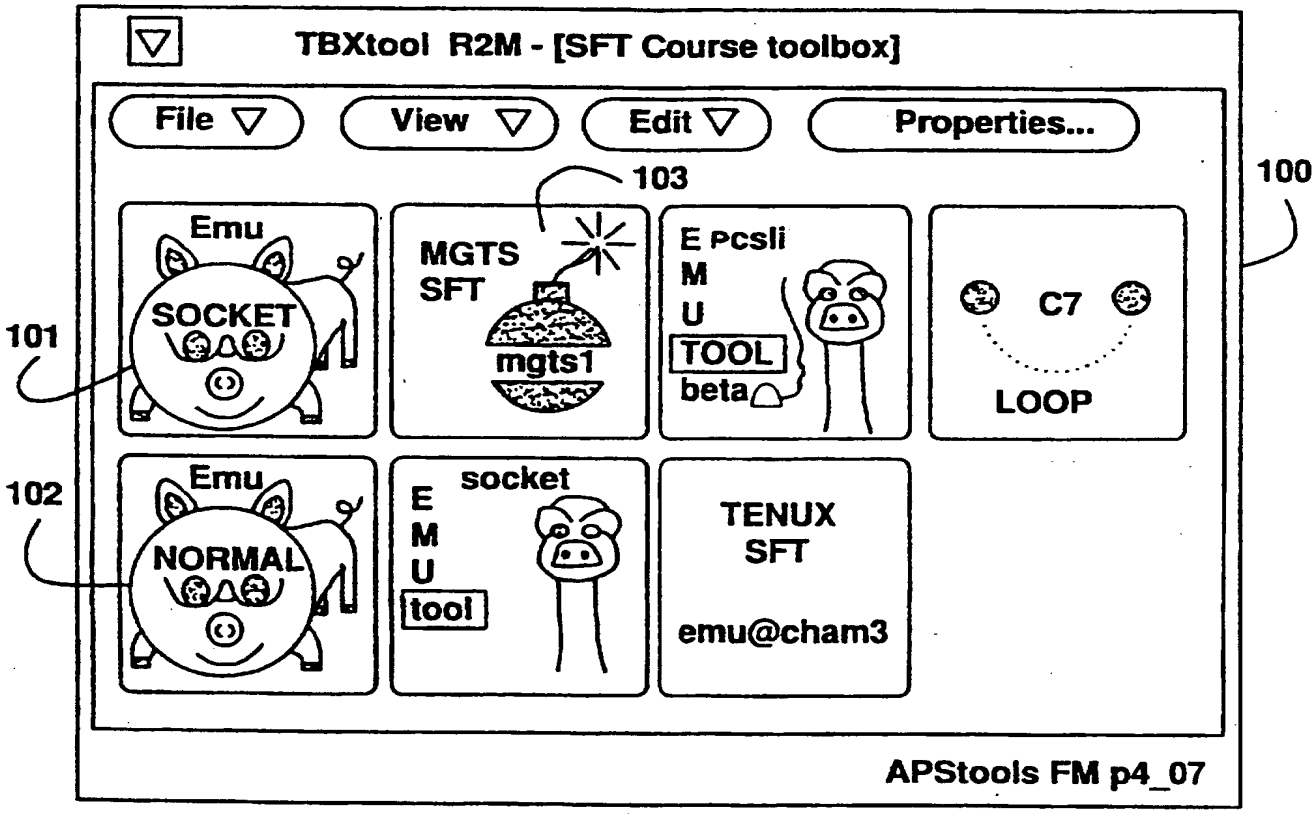

PIG - TOOL
CONVERTER
MESSAGE MONITOR
VIRTUAL PATCH PANEL
SIG. PT. - 1
SIG. PT. - 2
SIG. PT. - 3
SIG. PT. - n
93
84
88
EMULATOR
APPLICATION PROGRAMMING INTERFACE
95
APPLICATION SOFTWARE
EMULATED HARDWARE & SIGNALS
92
83
BACK END UNIX SOCKET
LAN
79
78
FRONT END UNIX SOCKET
PROTOCOL SIMULATOR
LIC - 1
LIC - 2
LIC - 3
LIC - n
91
PROTOCOL SIMULATION SOFTWARE
76
73

FIG.9

FIG.10



BNSDOCID: <WO___9735406A1_I_>

FIG.11

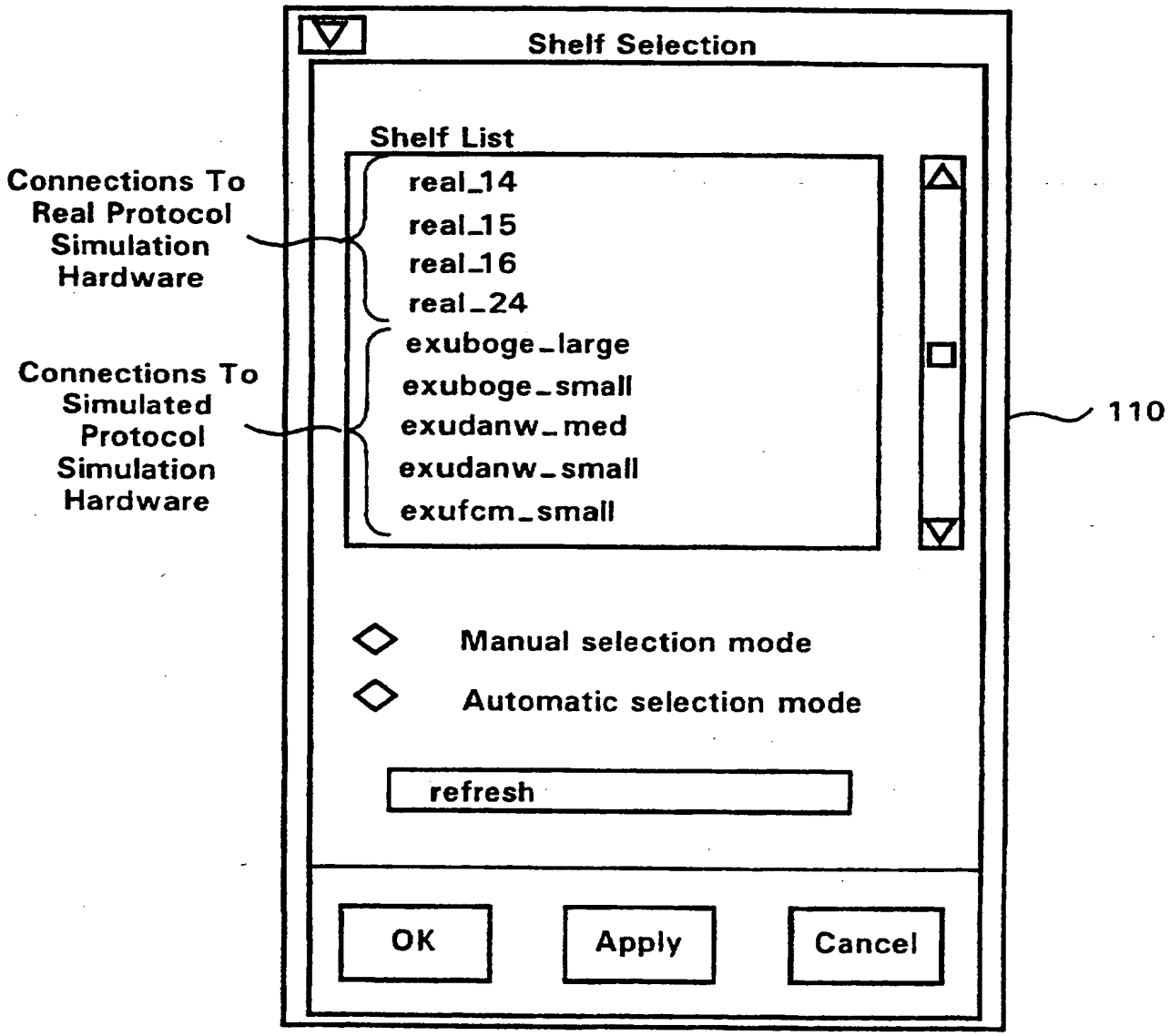
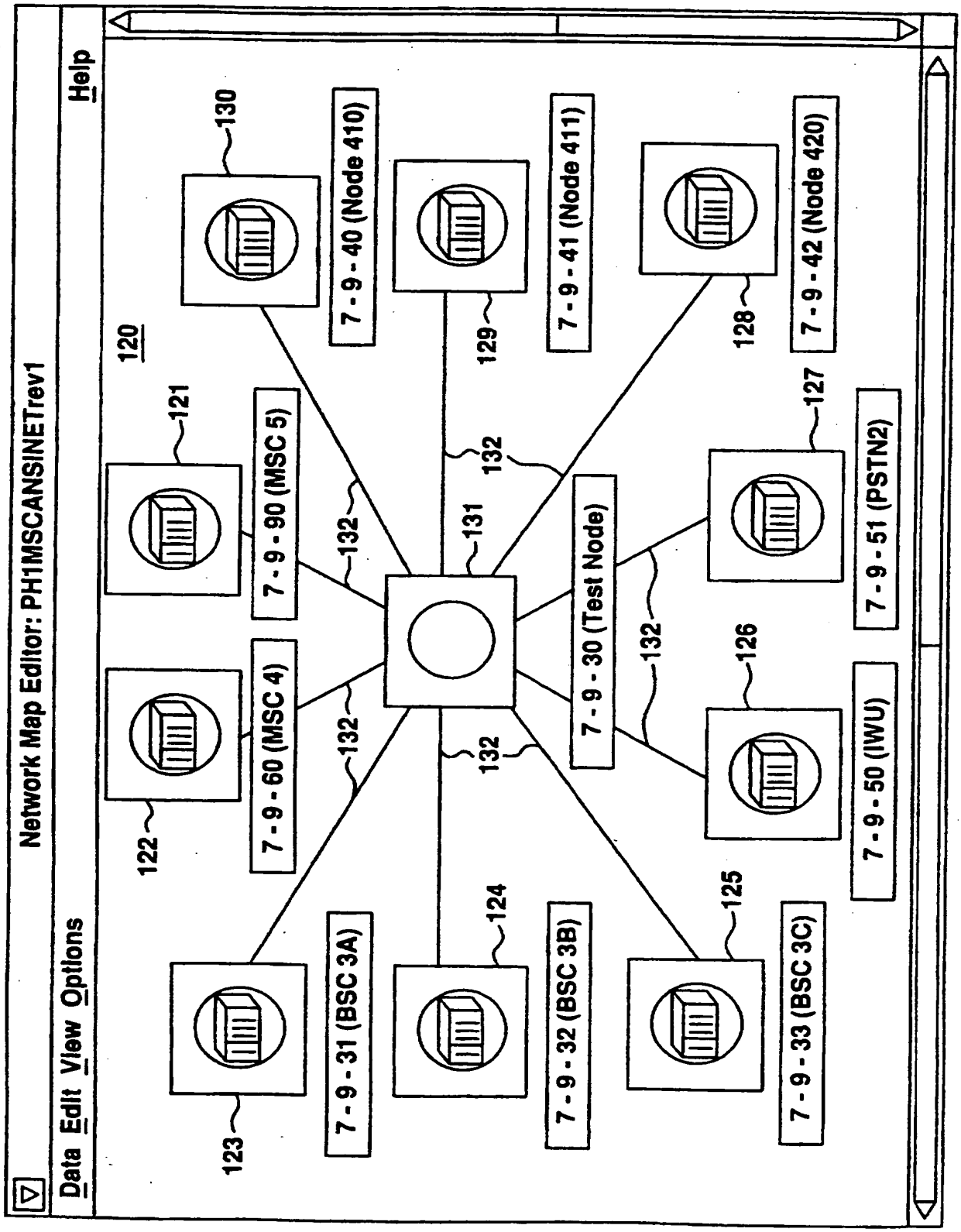

Shelf Selection
Shelf List
real_14
real_15
real_16
real_24
exuboge_large
exuboge_small
exudanw_med
exudanw_small
exufcm_small
Connections To Real Protocol Simulation Hardware
Connections To Simulated Protocol Simulation Hardware
110
Manual selection mode
Automatic selection mode
refresh
OK
Apply
Cancel

FIG.12



BNSDOCID: <WO___9735406A1_I_>

FIG.13

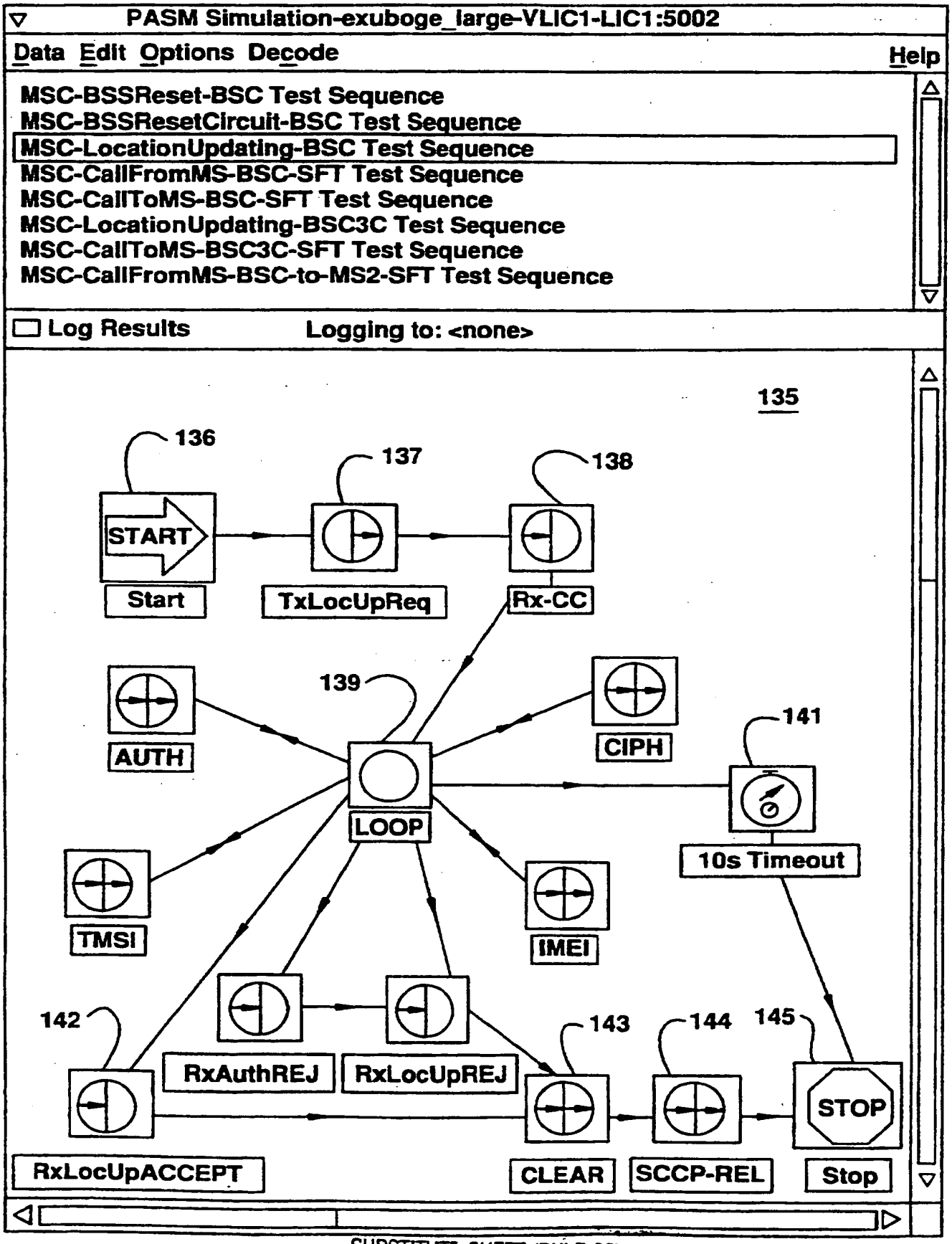
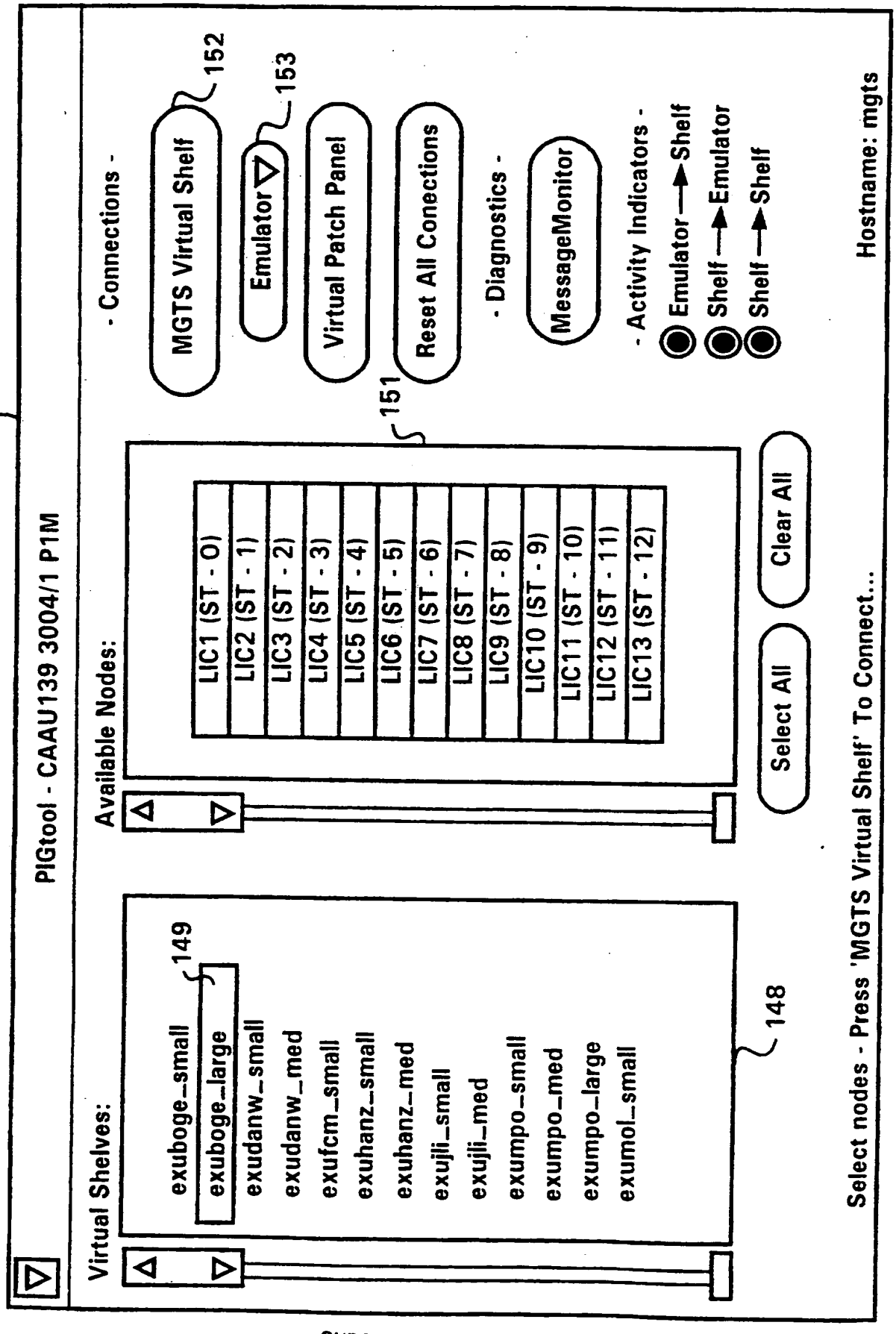
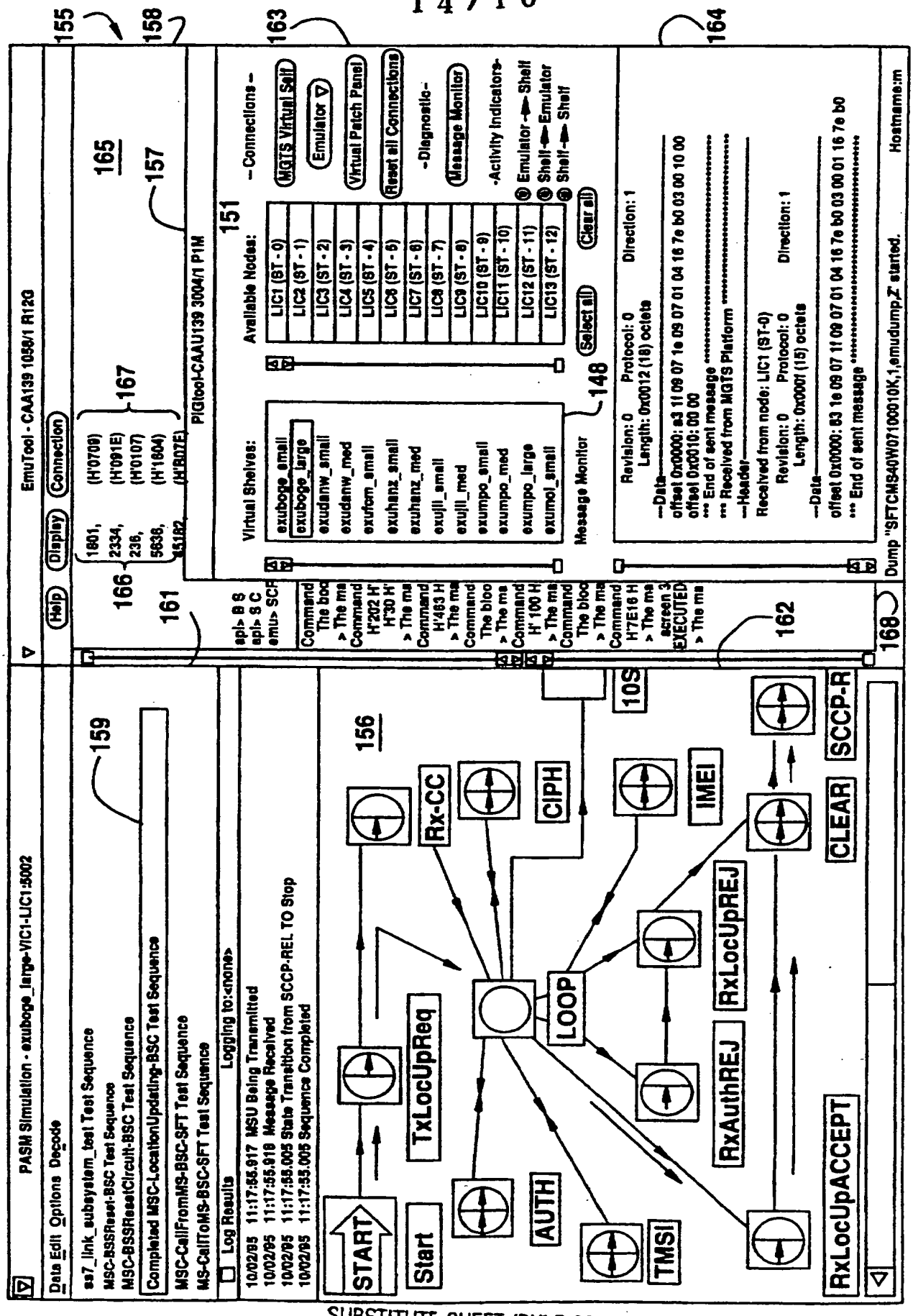

PASM Simulation-exuboge_large-VLIC1-LIC1:5002
Data Edit Options Decode
Help
MSC-BSSReset-BSC Test Sequence
MSC-BSSResetCircuit-BSC Test Sequence
MSC-LocationUpdating-BSC Test Sequence
MSC-CallFromMS-BSC-SFT Test Sequence
MSC-CallToMS-BSC-SFT Test Sequence
MSC-LocationUpdating-BSC3C Test Sequence
MSC-CallToMS-BSC3C-SFT Test Sequence
MSC-CallFromMS-BSC-to-MS2-SFT Test Sequence
Log Results
Logging to: <none>
135
136
137
138
START
Start
TxLocUpReq
Rx-CC
139
AUTH
CIPH
141
LOOP
10s Timeout
TMSI
IMEI
142
RxAuthREJ
RxLocUpREJ
143
144
145
STOP
RxLocUpACCEPT
CLEAR
SCCP-REL
Stop

FIG.14



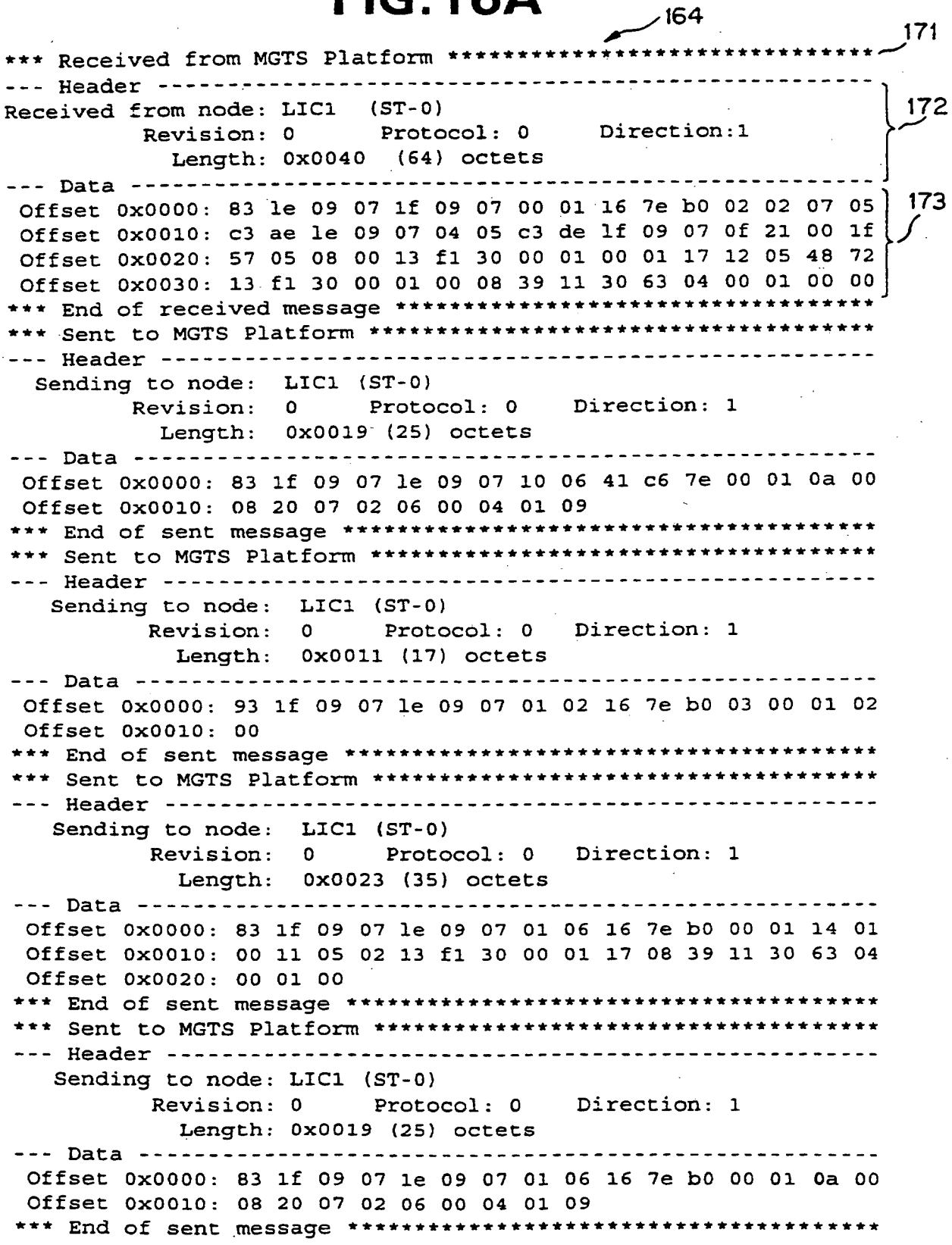
BNSDOCID: <WO__ 9735406A1_I_>

FIG.15



BNSDOCID: <WO 9735406A1 I >

FIG.16A



ENSDOCID: <WO__ 9735406A1_I_>

FIG.16B

164

```
*** Received from MGTS Platform *********************************
--- Header -----------------------------------------------------
Received from node: LIC1 (ST-0)
          Revision: 0     Protocol: 0    Direction: 1
            Length: 0x0012 (18) octets
--- Data -------------------------------------------------------
 Offset 0x0000: 83 1e 09 07 1f 09 07 00 06 03 00 01 00 01 03 00
 Offset 0x0010: 01 21
*** End of received message *************************************
*** Sent to MGTS Platform ***************************************
--- Header -----------------------------------------------------
   Sending to node: LIC1 (ST-0)
          Revision: 0     Protocol: 0    Direction: 1
            Length: 0x0012 (18) octets
--- Data -------------------------------------------------------
 Offset 0x0000: a3 1f 09 07 1e 09 07 01 04 16 7e b0 03 00 01 00
 Offset 0x0010: 00 00
*** End of sent message *****************************************
*** Received from MGTS Platform *********************************
--- Header -----------------------------------------------------
Received from node: LIC1 (ST-0)
          Revision: 0     Protocol: 0    Direction: 1
            Length: 0x000f (15) octets
--- Data -------------------------------------------------------
 Offset 0x0000: 83 1e 09 07 1f 09 07 00 05 03 00 01 16 7e b0
*** End of received message *************************************
```

INTERNATIONAL SEARCH REPORT

Interna l Application No
PCT/US 97/04658

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L12/26 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PROCEEDINGS OF THE TWENTY-NINTH HAWAÏ INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, vol. 1, 3 - 6 January 1996, WAILEA, HI, USA, pages 320-329, XP002036013 CHUN W J ET AL: "APPLYING DISTRIBUTED SIMULATION TO A COMMUNICATION PROTOCOL DEVELOPMENT ENVIRONMENT"	1,2,5-7, 16,17, 20-22
Y	see page 321, left-hand column, line 42 - page 322, right-hand column, line 34; figures 1,2 see page 322, right-hand column, line 35 - page 323, left-hand column, line 15; figure 4 see page 323, right-hand column, line 20 - page 326, left-hand column, line 4; figures 6,7	3,18

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

3

Date of the actual completion of the international search	Date of mailing of the international search report
24 July 1997	13.08.1997

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Dupuis, H

Form PCT/ISA/210 (second sheet) (July 1992)

BNSDOCID: <WO__9735406A1_I_>

INTERNATIONAL SEARCH REPORT

Internat. Application No
PCT/US 97/04658

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MICROPROCESSING AND MICROPROGRAMMING, vol. 40, no. 10/12, 1 December 1994, pages 703-705, XP000504946 MISHOLCZI J ET AL: "CONFORMANCE TESTING OF X.25 PACKET LEVEL"	1,2,5,6, 16,17, 20,21
Y	see the whole document	3,18

X	DE 39 07 237 A (RUTENBECK WILHELM GMBH & CO ;INTEC GES FUER INFORMATIONSTEC (DE)) 13 September 1990 see column 1, line 1-13 see column 1, line 43-53 see column 2, line 20-54 see column 3, line 8-12	1,4,8, 11,16, 19,23,26

A	ELECTRICAL COMMUNICATION, vol. 65, no. 1, 1 January 1991, pages 60-66, XP000264670 "TELECOMMUNICATION TEST SYSTEMS" see page 62, left-hand column, line 17-58 see page 63, left-hand column, line 10-27 see page 64, right-hand column, line 29-43	1,8,16, 23

3

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

BNSDOCID <WO 9735406A1 I >

INTERNATIONAL SEARCH REPORT

Information on patent family members

Internat Application No
PCT/US 97/04658

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
DE 3907237 A	13-09-90	NONE	

BNSDOCID: <WO__9735406A1_I_>

FIG. 1
PRIOR ART

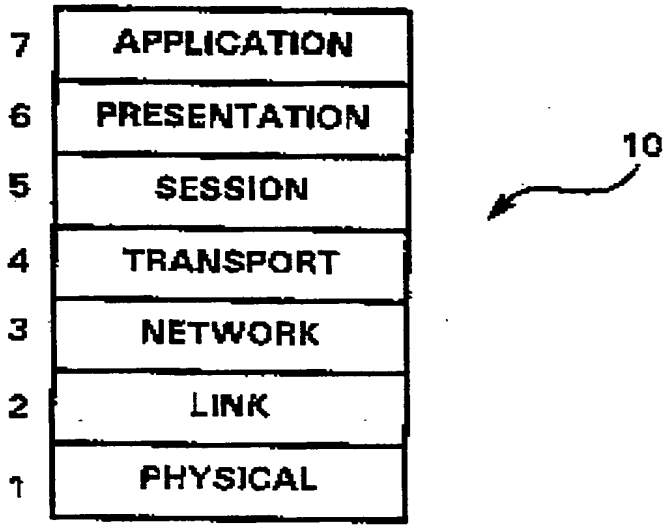


FIG. 2

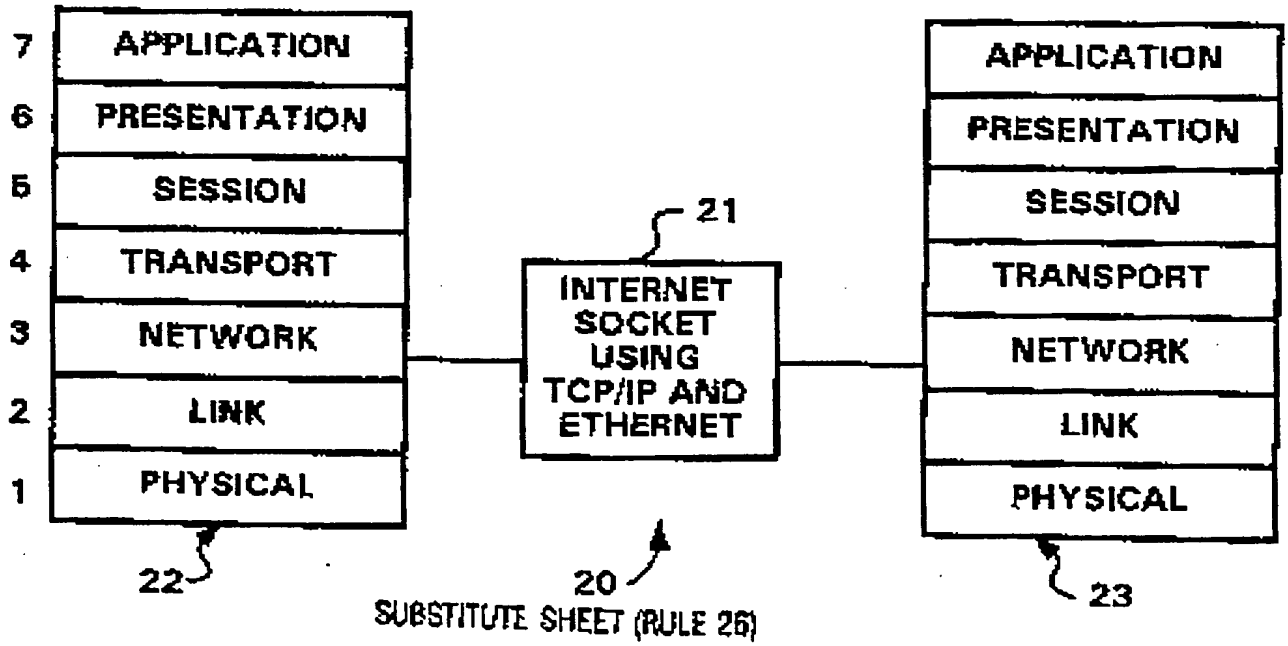
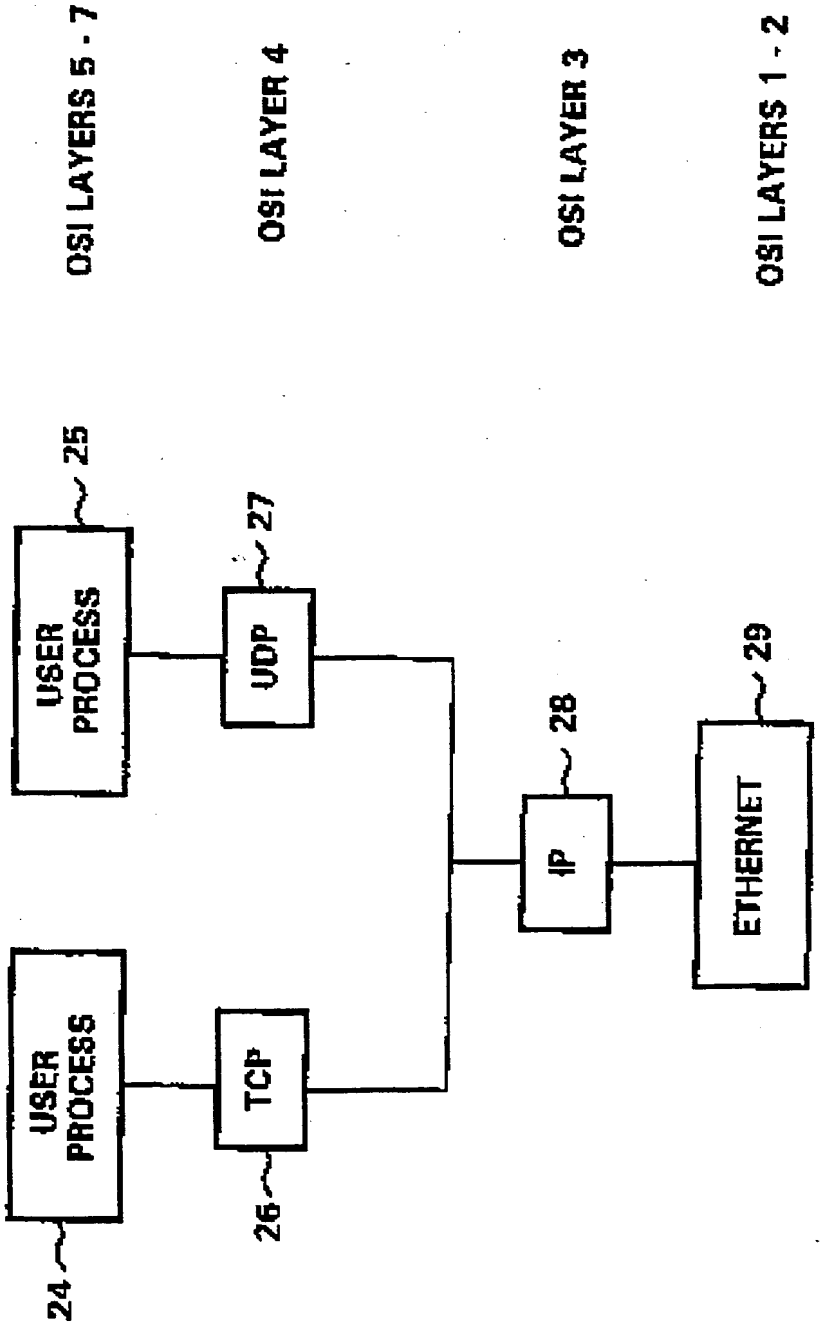


FIG.2A



BNSDOCID: <WO___9735406A1TI_>

FIG.3

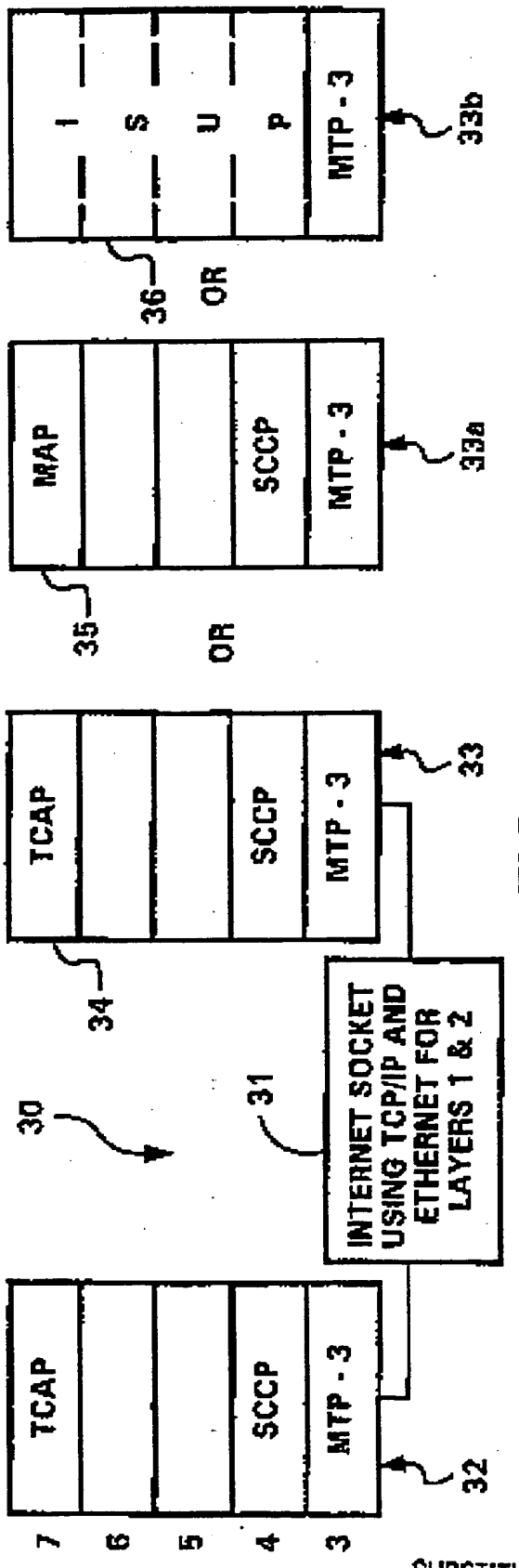

I
S
U
P
MTP - 3
33b
36
OR
MAP
SCCP
MTP - 3
33a
35
OR
TCAP
SCCP
MTP - 3
33
34
30
31
INTERNET SOCKET USING TCP/IP AND ETHERNET FOR LAYERS 1 & 2
TCAP
SCCP
MTP - 3
32
7
6
5
4
3

FIG.4

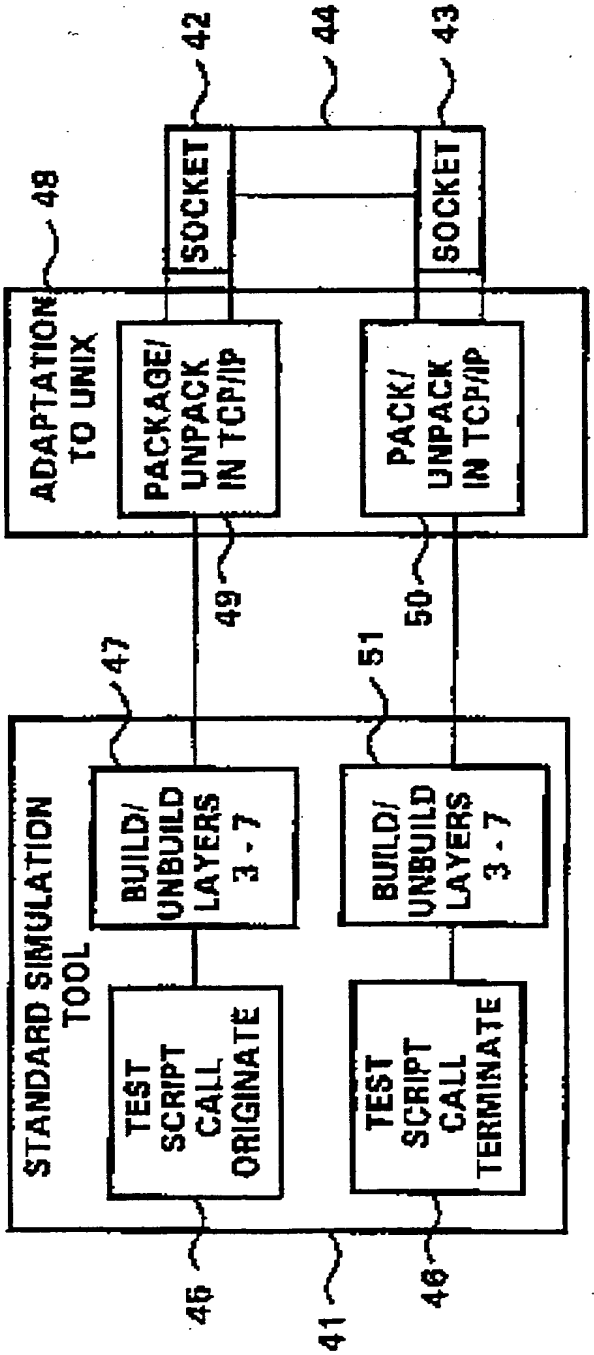
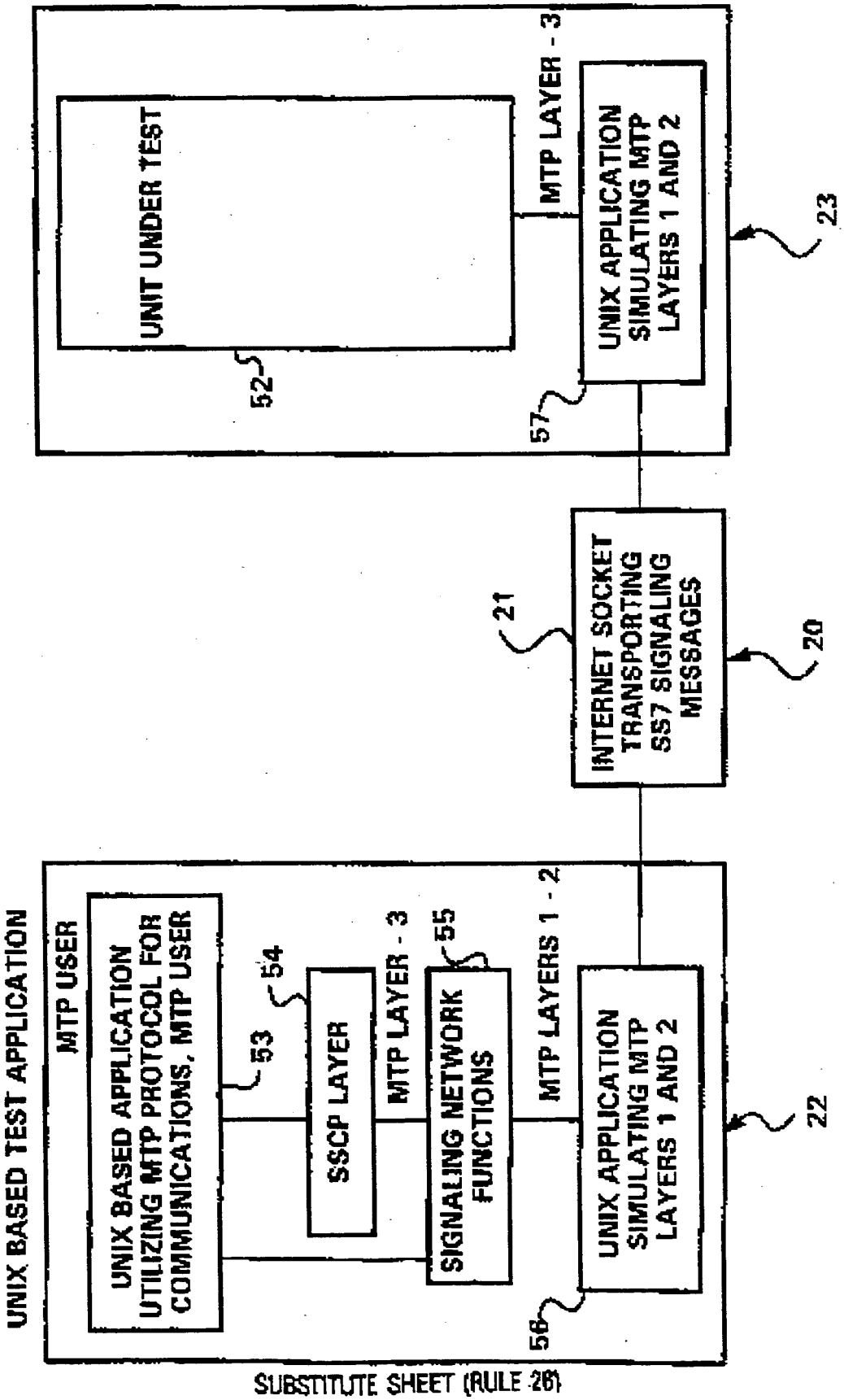

42
44
43
48
SOCKET
SOCKET
ADAPTATION TO UNIX
PACKAGE/ UNPACK IN TCP/IP
PACK/ UNPACK IN TCP/IP
49
50
47
51
STANDARD SIMULATION TOOL
BUILD/ UNBUILD LAYERS 3 - 7
BUILD/ UNBUILD LAYERS 3 - 7
TEST SCRIPT CALL ORIGINATE
TEST SCRIPT CALL TERMINATE
45
41
46

FIG.5



ENSDOCID: <WO__ 9735406A1TI >

FIG.6

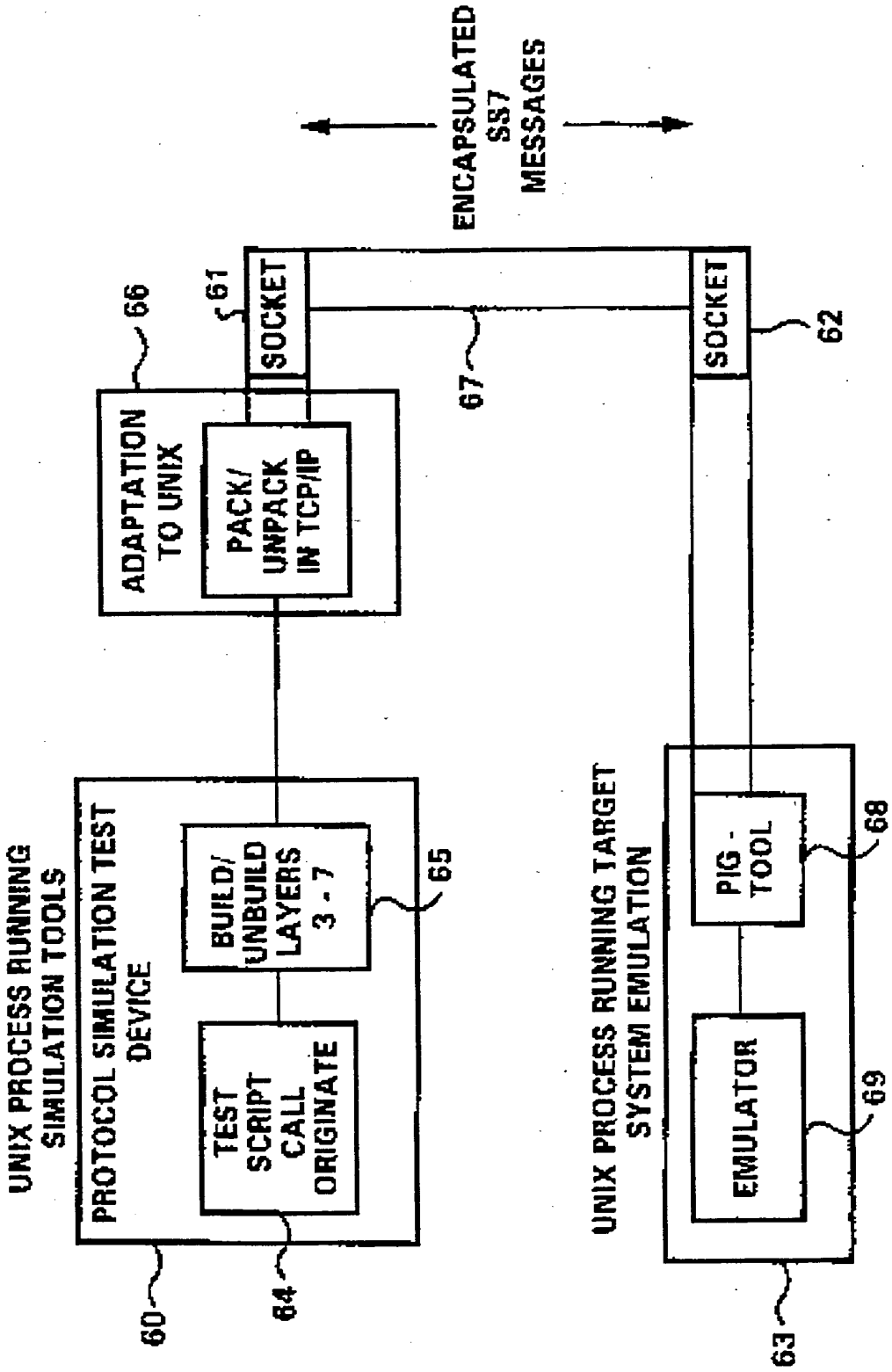
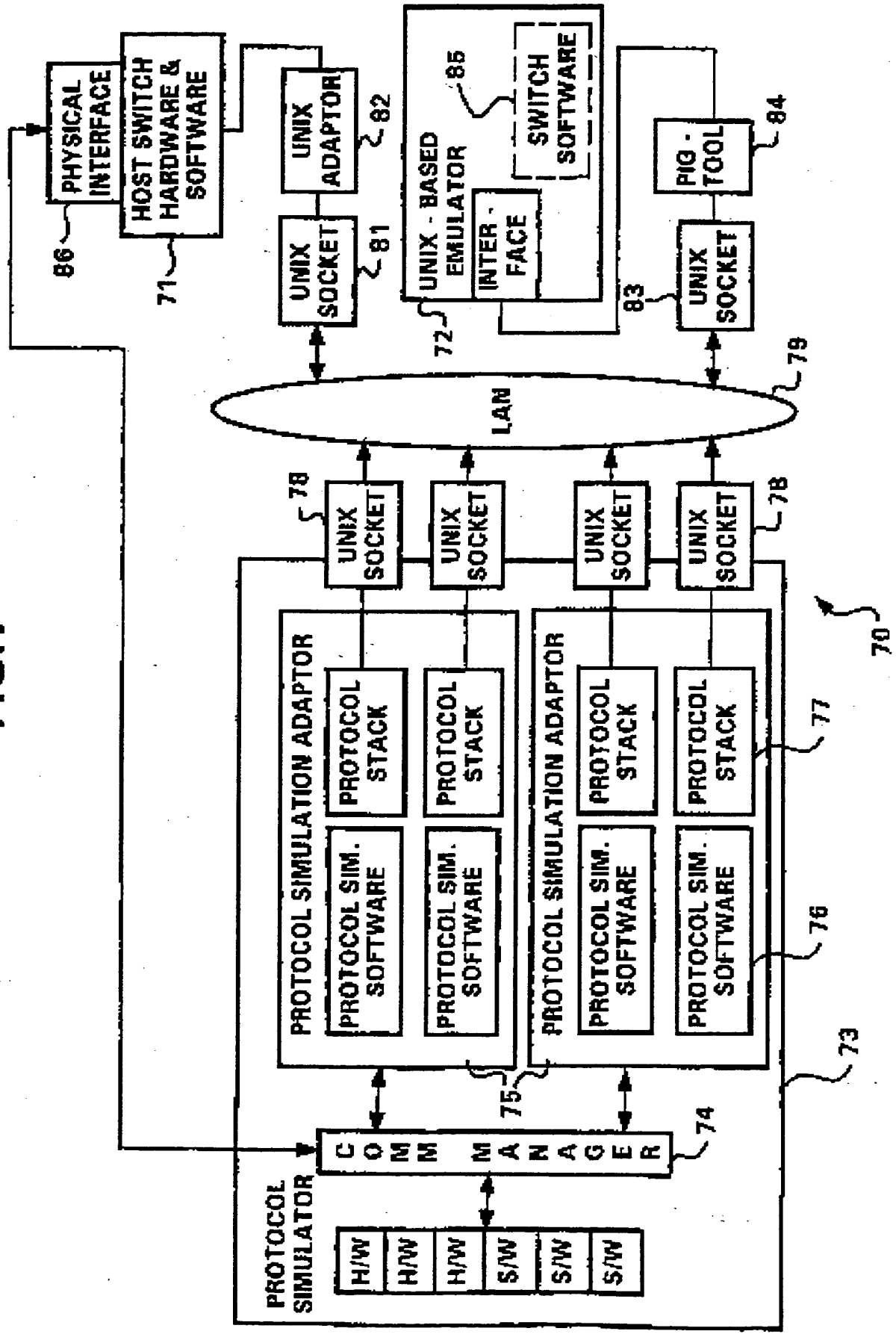
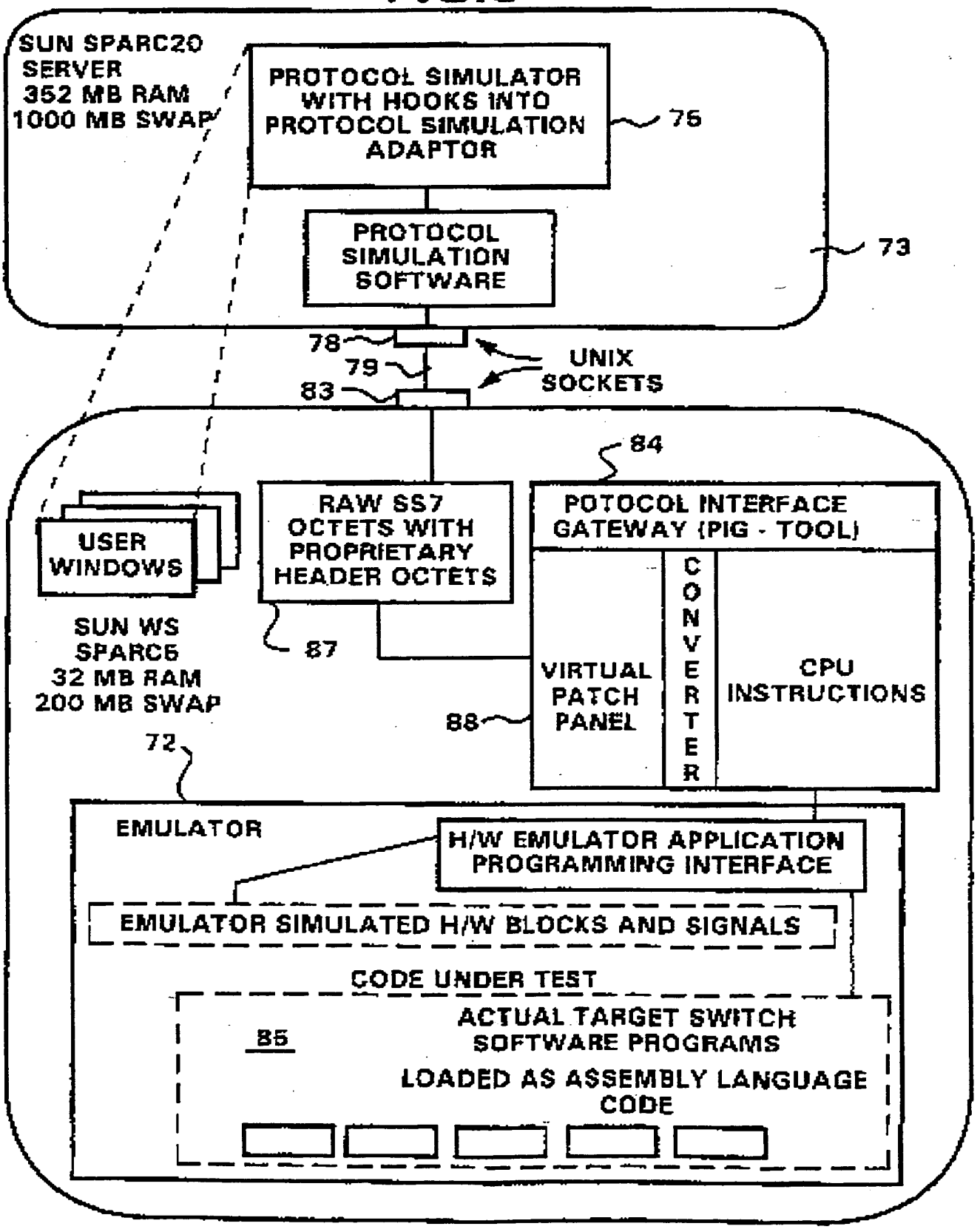

ENCAPSULATED SS7 MESSAGES
SOCKET
SOCKET
61
62
67
66
ADAPTATION TO UNIX
PACK/ UNPACK IN TCP/IP
UNIX PROCESS RUNNING SIMULATION TOOLS
PROTOCOL SIMULATION TEST DEVICE
BUILD/ UNBUILD LAYERS 3 - 7
TEST SCRIPT CALL ORIGINATE
65
60
64
UNIX PROCESS RUNNING TARGET SYSTEM EMULATION
PIG - TOOL
EMULATOR
68
69
63

FIG.7



ENSDOCID: <WO___9735406A1TI_>

FIG.8



BNSDOCID: <WO__9735406A1TI_>

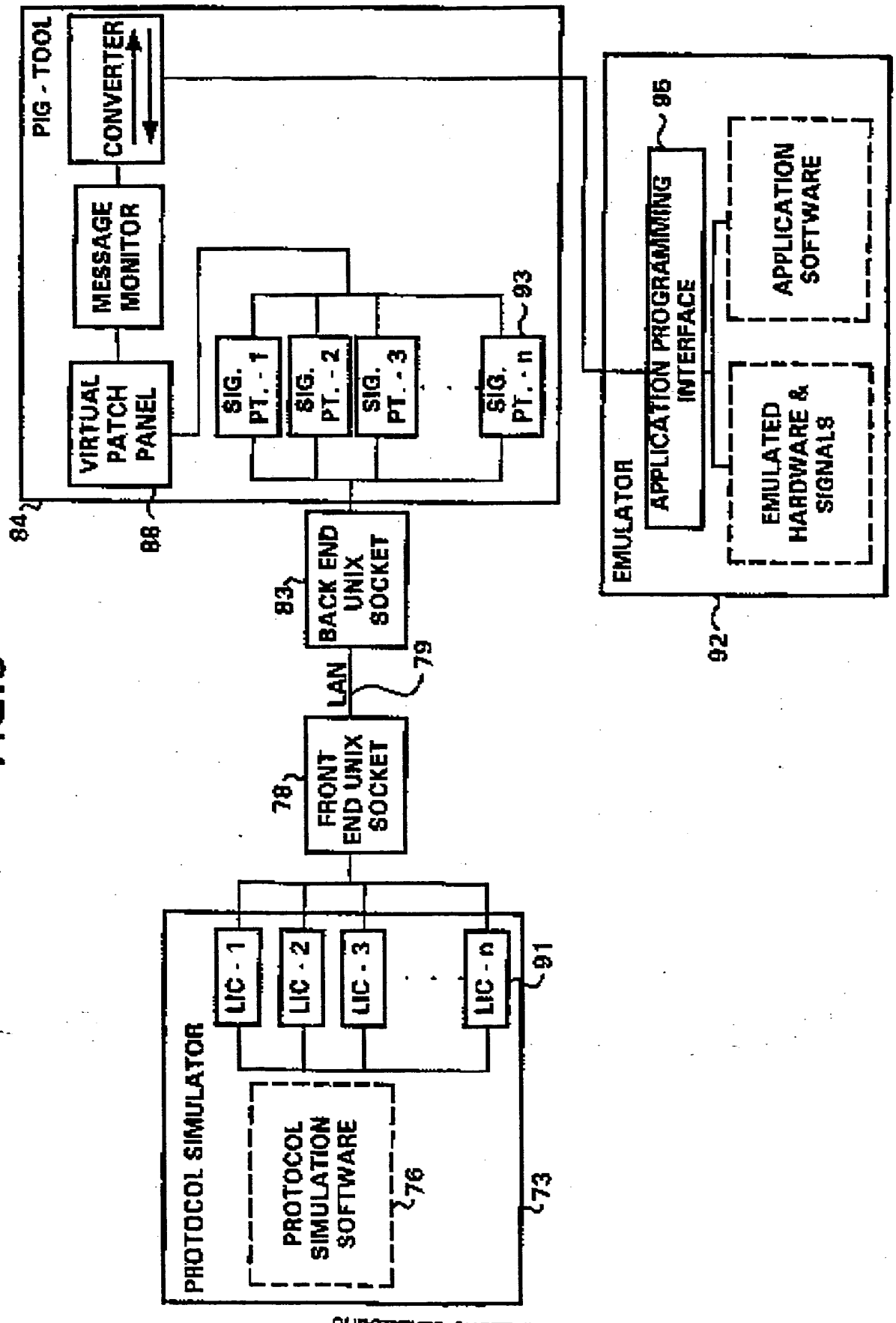
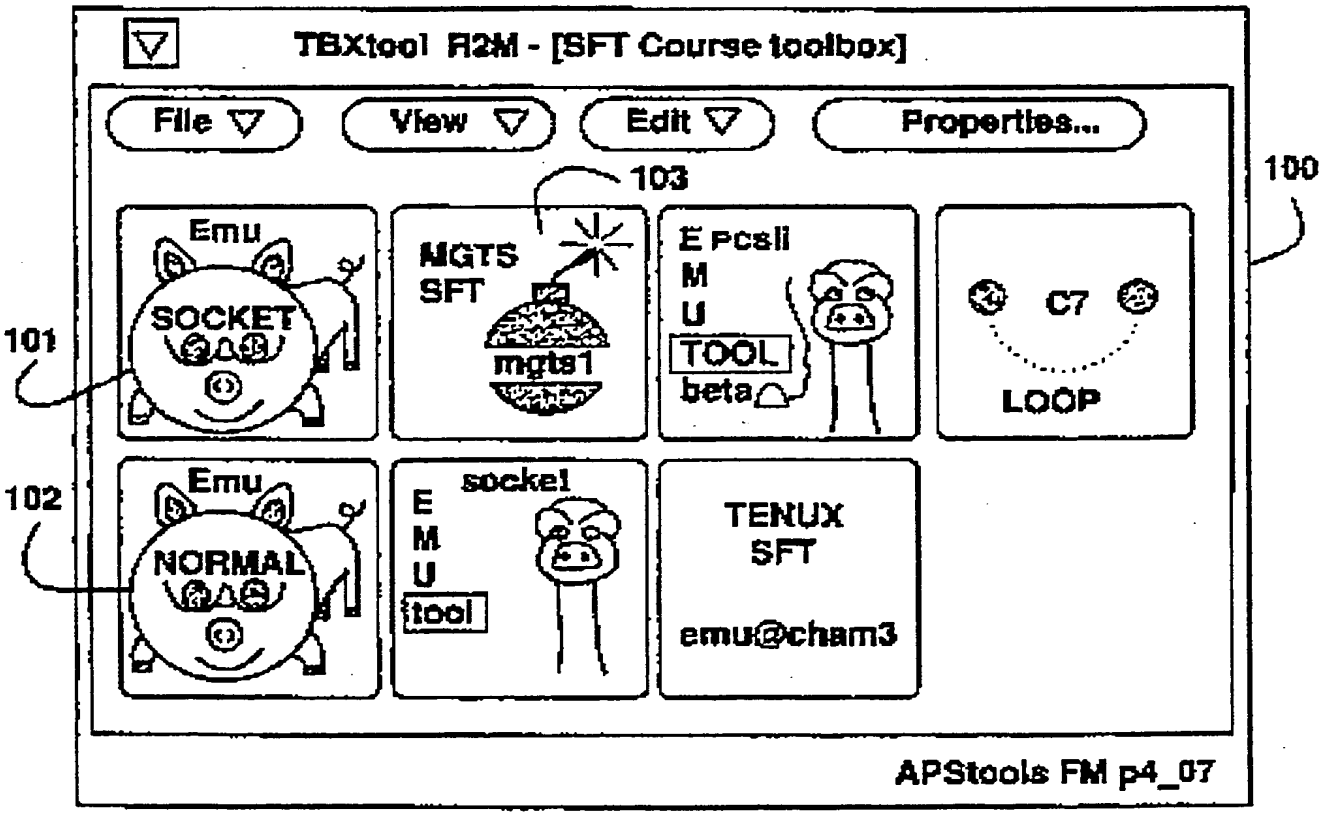


FIG.9

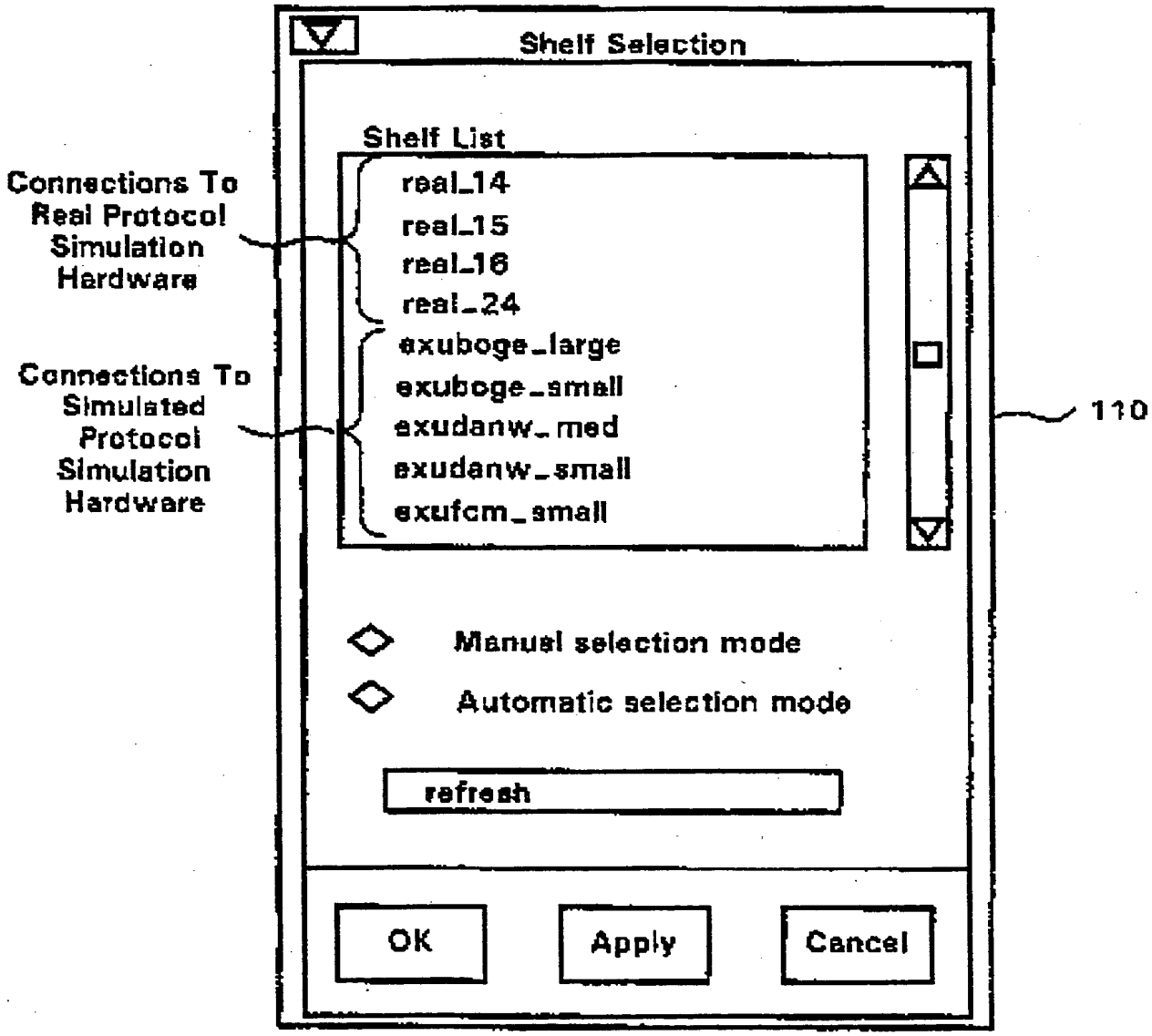
BNSDOCID: <WO_ 9735406A1TI_>

FIG. 10



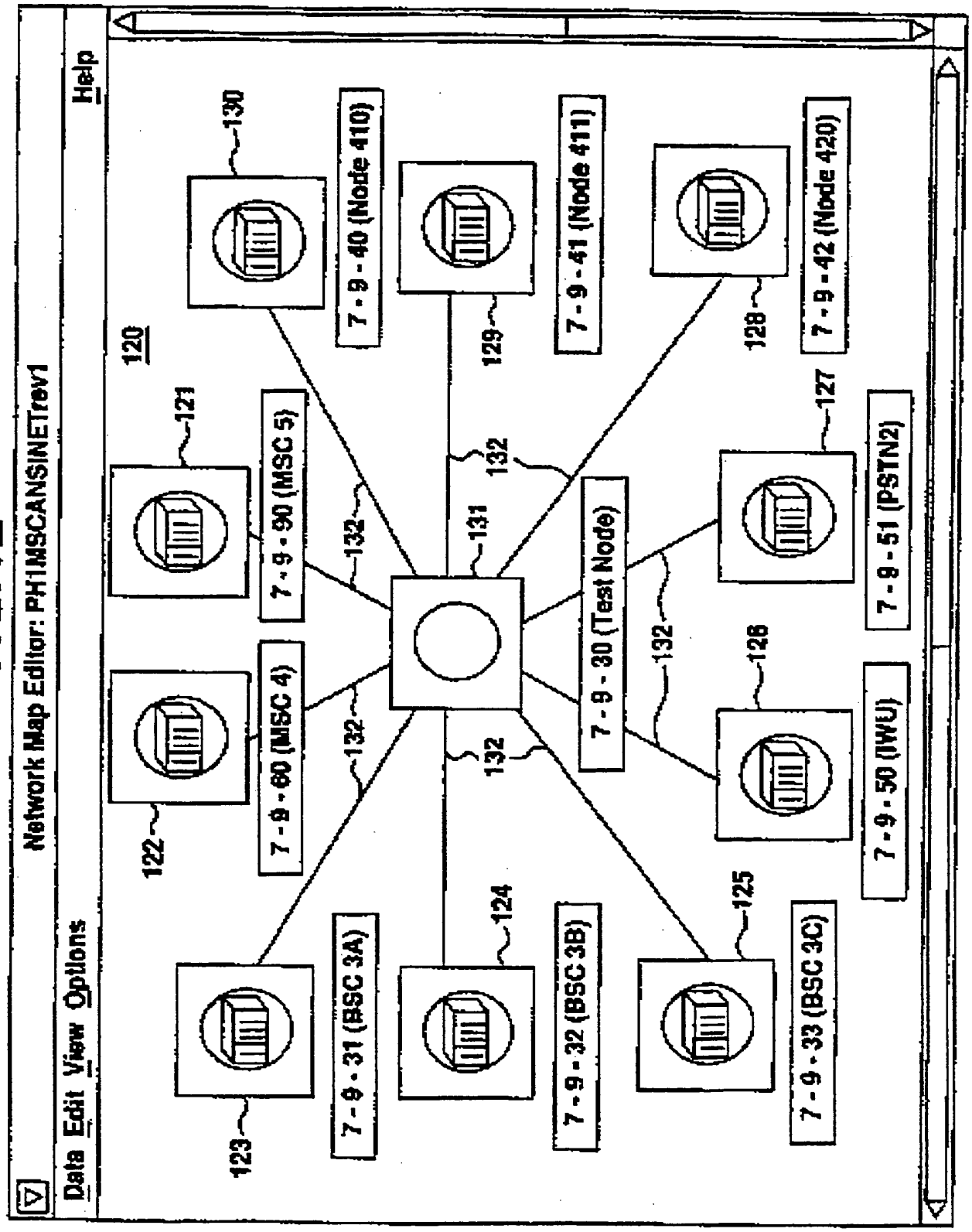
BNSDOCID: <WO _ 9735406A1TI >

FIG.11



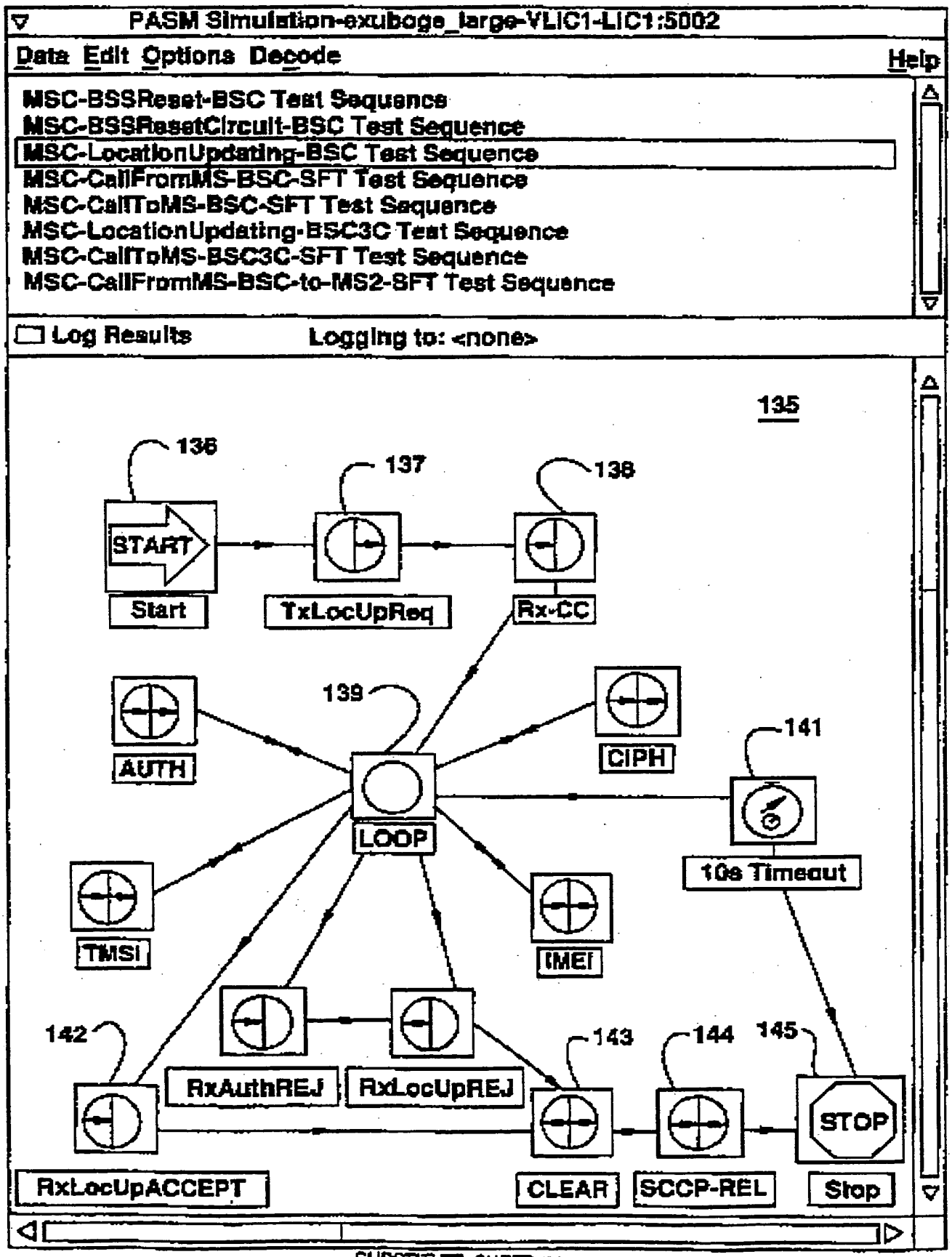
BNSDOCID: <WO_ _9735406A1TI >

FIG. 12



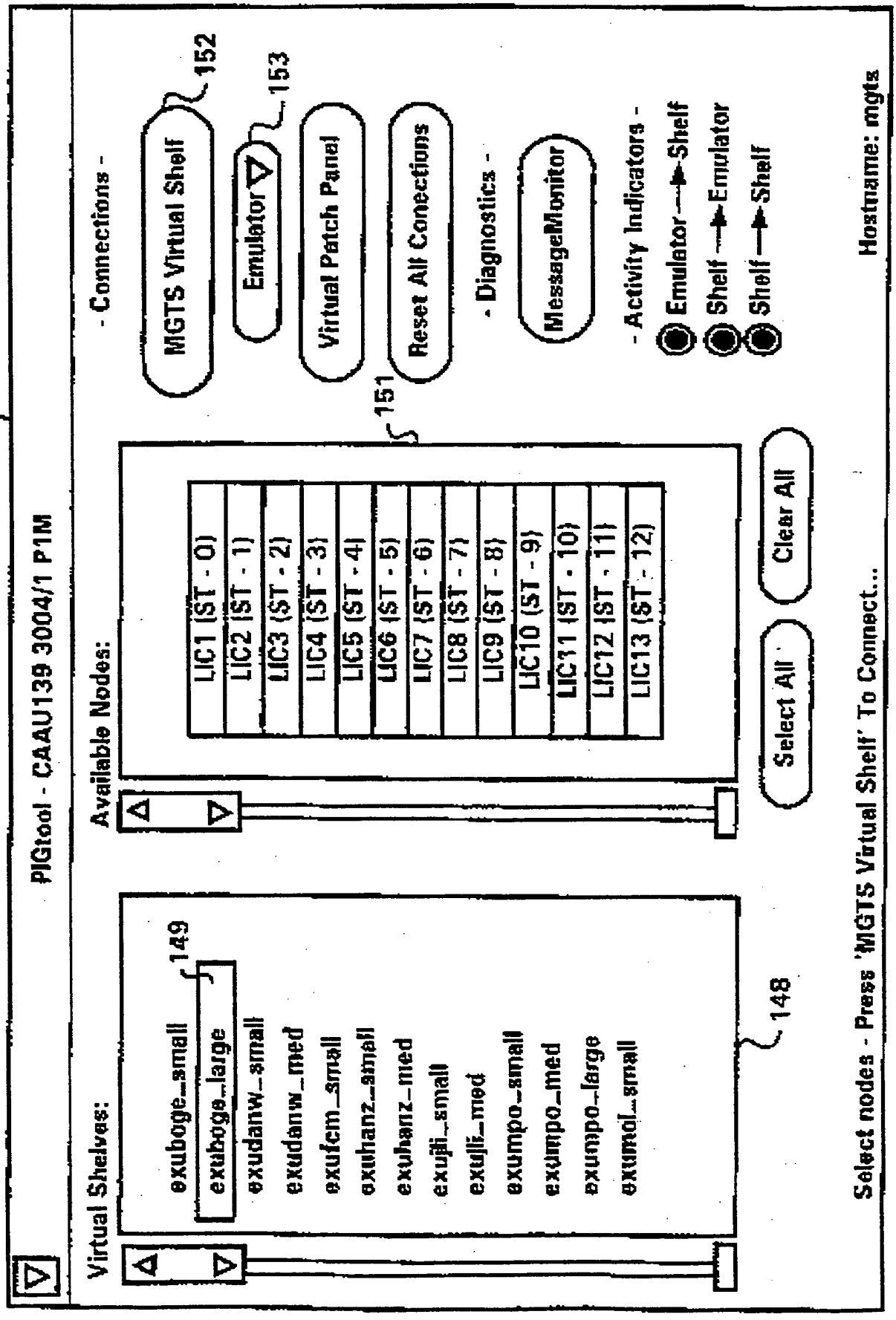
ENSDOCID: <WO_ 9735406A1TI >

FIG.13



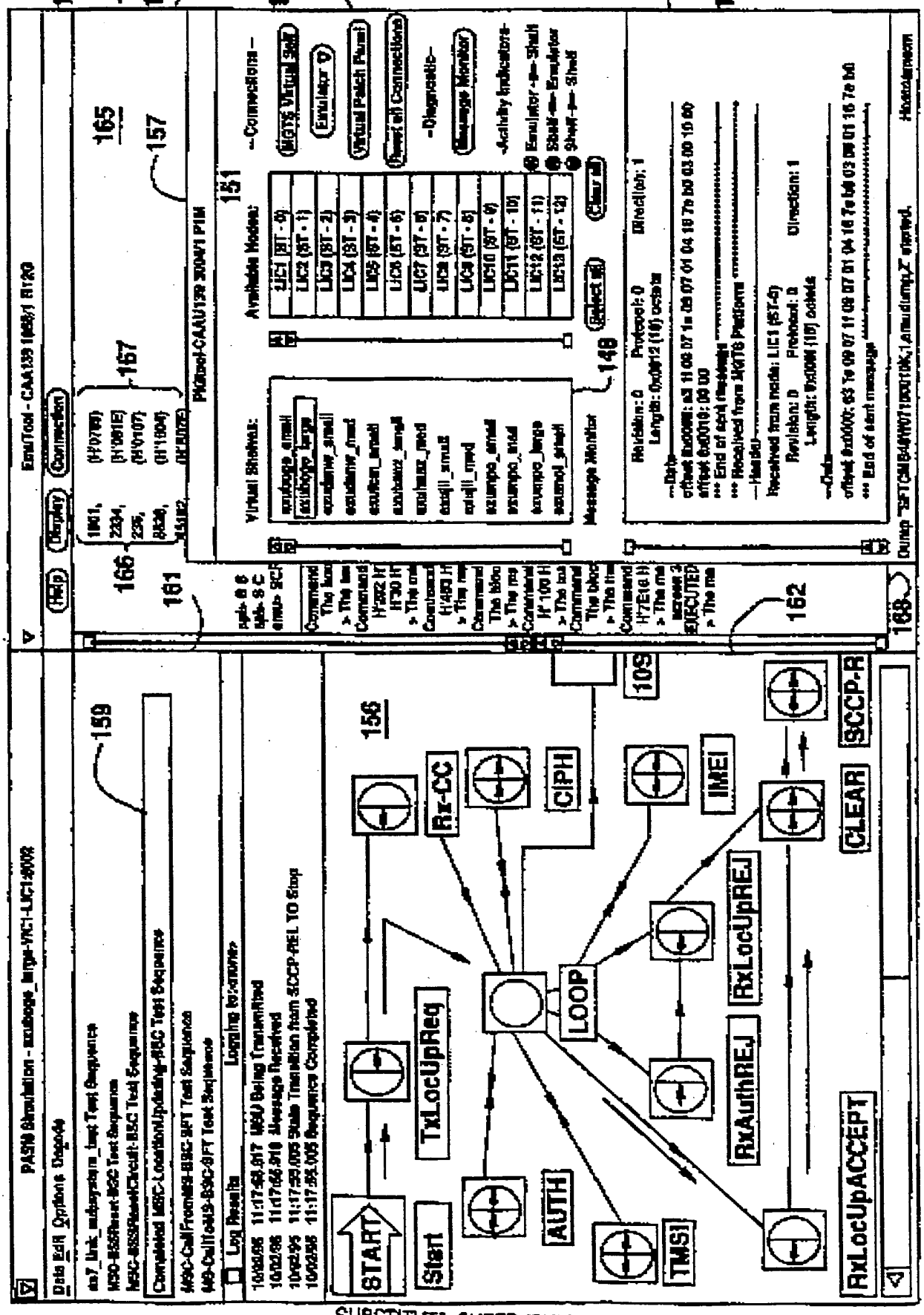
BNSDOCID: <WO_ _9735406A1TI_>

FIG.14



BNSDOCID: <WO 9735406A1TI >

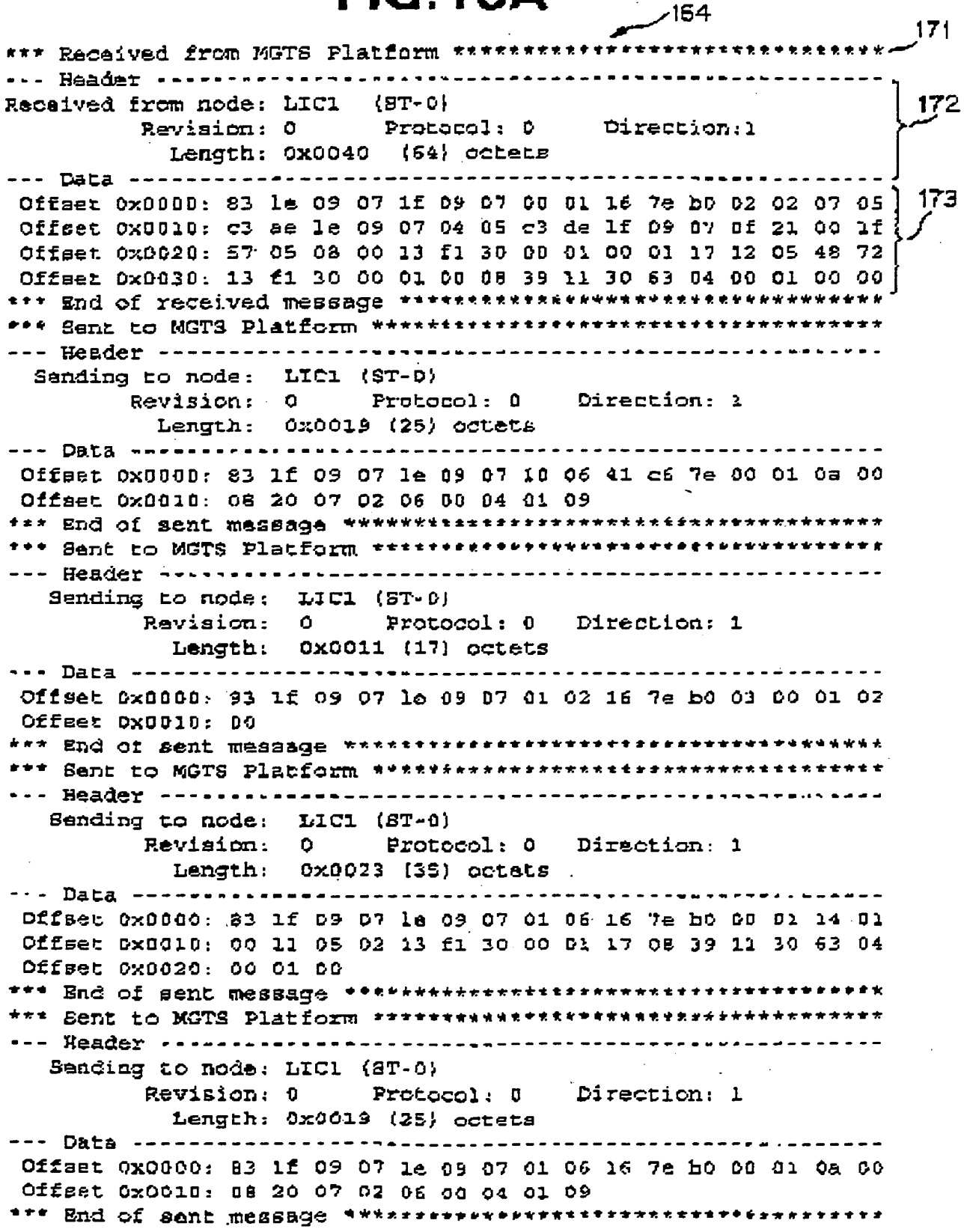
FIG.15



SUBSTITUTE SHEET (RULE 26)

BNSDOCID: <WO___ 9735406A1TI_>

FIG.16A



BNSDOCID: <WO__9735406A1TI_>

FIG. 16B

164

```
*** Received from MGTS Platform ********************************
--- Header -----------------------------------------------------
Received from node: LIC1 (ST-0)
          Revision: 0     Protocol: 0    Direction: 1
            Length: 0x0012 (18) octets
--- Data -------------------------------------------------------
 Offset 0x0000: 83 1e 09 07 1f 09 07 00 06 03 00 01 00 01 03 00
 Offset 0x0010: 01 21
*** End of received message ***************************************
*** Sent to MGTS Platform *****************************************
--- Header -----------------------------------------------------
   Sending to node: LIC1 (ST-0)
          Revision: 0     Protocol: 0    Direction: 1
            Length: 0x0012 (18) octets
--- Data -------------------------------------------------------
 Offset 0x0000: a3 1f 09 07 1e 09 07 01 04 16 7e b0 03 00 01 00
 Offset 0x0010: 00 00
*** End of sent message *******************************************
*** Received from MGTS Platform ***********************************
--- Header -----------------------------------------------------
Received from node: LIC1 (ST-0)
          Revision: 0     Protocol: 0    Direction: 1
            Length: 0x000f (15) octets
--- Data -------------------------------------------------------
 Offset 0x000D: 83 1e 09 07 1f 09 07 00 05 03 00 01 16 7e b0
*** End of received message ***************************************
```


BNSDOCID: <WO_ _9735406A1TI_>